Privater Masterstudiengang
Programmierung von Grund auf

tech global university

tech global university

Privater Masterstudiengang
Programmierung von Grund auf

- Modalität: **online**
- Dauer: **12 Monate**
- Qualifizierung: **TECH Global University**
- Akkreditierung: **60 ECTS**
- Zeitplan: **in Ihrem eigenen Tempo**
- Prüfungen: **online**

Internetzugang: www.techtitute.com/de/informatik/masterstudiengang/masterstudiengang-programmierung-grund

Index

01 Präsentation des Programms — Seite 4

02 Warum an der TECH studieren? — Seite 8

03 Lehrplan — Seite 12

04 Lehrziele — Seite 24

05 Karrieremöglichkeiten — Seite 30

06 Studienmethodik — Seite 34

07 Lehrkörper — Seite 44

08 Qualifizierung — Seite 48

01

Präsentation des Programms

Programmieren ist im 21. Jahrhundert zu einer unverzichtbaren Fähigkeit geworden, die eine Schlüsselrolle bei der digitalen Transformation von Gesellschaften und Volkswirtschaften spielt. Einem OECD-Bericht zufolge sind für mehr als 60% der heutigen Arbeitsplätze fortgeschrittene digitale Fähigkeiten erforderlich, von denen die Programmierung eine der am meisten gefragten ist. In Anbetracht der Tatsache, dass es sich hierbei um einen im digitalen Zeitalter unverzichtbaren Bereich handelt, hat TECH dieses Programm entwickelt, das sich mit den Grundlagen dieser Disziplin und ihrer Anwendung in weltweit verbreiteten Sprachen befasst. Mit Hilfe einer 100%igen Online-Methodik lernen die Spezialisten, Algorithmen zu strukturieren, Daten zu verwalten und funktionale Projekte zu entwickeln, und schaffen sich so eine solide Grundlage, um in der technologischen Welt weiter voranzukommen.

Programmieren war noch nie so einfach zu erlernen. Mit dem privaten Masterstudiengang von TECH werden Sie dank der 100%igen Online-Methodik von überall aus Sprachen wie Python und JavaScript beherrschen. Schreiben Sie sich jetzt ein!"

Die Programmierung ist das Herzstück der digitalen Transformation, die moderne Gesellschaften durchlaufen. Von der Entwicklung mobiler Anwendungen bis zur Automatisierung von Geschäftsprozessen ist die Programmierkenntnis eine grundlegende Fähigkeit, um sich in einem sich ständig weiterentwickelnden Arbeitsumfeld zurechtzufinden. In diesem Sinne werden Fachleute, die dieses Gebiet beherrschen, nicht nur bessere Wachstumschancen haben, sondern auch die Fähigkeit verbessern, sich in einer Welt, in der sich die Technologie ständig weiterentwickelt, anzupassen und erfolgreich zu sein.

In Anbetracht dieser Notwendigkeit bietet TECH einen Privaten Masterstudiengang in Programmierung von Grund auf an, ein umfassendes Programm, das die neuesten Kenntnisse in diesem Bereich vermittelt. Der von Branchenexperten entwickelte Lehrplan reicht von den Grundlagen von Algorithmen und Datenstrukturen bis hin zur praktischen Anwendung von Sprachen wie *Python*, *JavaScript* und *HTML*. Außerdem wird es Module geben, die auf Webdesign, Anwendungsentwicklung und Datenbankmanagement spezialisiert sind. Auf diese Weise erwerben die Spezialisten technisches Wissen und entwickeln analytische und kreative Fähigkeiten, um Probleme durch Programmierung zu lösen. All dies mit einem praktischen Ansatz, der die Umsetzung der gelernten Konzepte gewährleistet.

Durch die Spezialisierung auf diesen Bereich werden die Absolventen nicht nur neue Möglichkeiten im Technologiesektor finden, sondern auch in Branchen wie Finanzen, Gesundheitswesen oder Logistik, in denen die Programmierung eine Schlüsselrolle spielt, herausragende Leistungen erbringen. Darüber hinaus werden sie für die Leitung von Entwicklungsprojekten, die Automatisierung von Prozessen und die Optimierung von Ressourcen qualifiziert, wodurch ihr berufliches Profil verbessert und ihr Zugang zu hochgradig wettbewerbsfähigen Beschäftigungsmöglichkeiten erleichtert wird.

Schließlich bietet die 100%ige Online-Variante dieses Programms die nötige Flexibilität, um das Studium mit anderen persönlichen oder beruflichen Verpflichtungen in Einklang zu bringen. TECH wird den Studenten eine innovative Plattform zur Verfügung stellen, die rund um die Uhr zugänglich ist. Darüber hinaus wird die *Relearning*-Methode die Fortbildung durch die Wiederholung der wichtigsten Konzepte in verschiedenen Kontexten optimieren, was eine schrittweise und effiziente Aufnahme des Inhalts ermöglicht.

Dieser **Privater Masterstudiengang in Programmierung von Grund auf** enthält das vollständigste und aktuellste Programm auf dem Markt. Seine herausragendsten Merkmale sind:

- Die Entwicklung von Fallstudien, die von Experten für Programmierung präsentiert werden
- Der anschauliche, schematische und äußerst praxisnahe Inhalt vermittelt alle für die berufliche Praxis unverzichtbaren wissenschaftlichen und praktischen Informationen
- Er enthält praktische Übungen, in denen der Selbstbewertungsprozess durchgeführt werden kann, um das Lernen zu verbessern
- Sein besonderer Schwerpunkt liegt auf innovativen Methoden
- Theoretische Lektionen, Fragen an den Experten, Diskussionsforen zu kontroversen Themen und individuelle Reflexionsarbeit
- Die Verfügbarkeit des Zugriffs auf die Inhalte von jedem festen oder tragbaren Gerät mit Internetanschluss

Von grundlegenden Algorithmen bis hin zur Anwendungsentwicklung – dieses Programm wird Sie in kurzer Zeit vom Anfänger zum Experten machen. Sie werden zu dem Profi, den die Technologiebranche braucht!“

Tauchen Sie mit einem unvergleichlichen Lehrplan in die Welt der Programmierung ein. Mit dem umfassendsten privaten Masterstudiengang auf dem Markt werden Sie Python, JavaScript und Webentwicklung von Grund auf beherrschen"

Unterstützt von technologisch versierten Dozenten führt Sie dieser Abschluss Schritt für Schritt zur Beherrschung der gefragtesten Programmiersprachen auf dem Markt. Beginnen Sie noch heute und gestalten Sie eine bessere Zukunft!

In diesem privaten Masterstudiengang werden Sie einen innovativen Ansatz entdecken, der Theorie und reale Projekte kombiniert. Schreiben Sie sich jetzt ein und erwecken Sie Ihre Ideen mit Code zum Leben!

Zu den Dozenten gehören Fachleute aus dem Bereich der Programmierung, die ihre Erfahrungen in dieses Programm einbringen, sowie anerkannte Spezialisten aus führenden Gesellschaften und renommierten Universitäten.

Die multimedialen Inhalte, die mit den neuesten Bildungstechnologien entwickelt wurden, ermöglichen der Fachkraft ein situiertes und kontextbezogenes Lernen, d. h. eine simulierte Umgebung, die eine immersive Fortbildung bietet, die auf die Ausführung von realen Situationen ausgerichtet ist.

Das Konzept dieses Programms konzentriert sich auf problemorientiertes Lernen, bei dem der Student versuchen muss, die verschiedenen Situationen aus der beruflichen Praxis zu lösen, die während des gesamten Studiengangs gestellt werden. Dabei wird die Fachkraft durch ein innovatives interaktives Videosystem unterstützt, das von anerkannten Experten entwickelt wurde.

02

Warum an der TECH studieren?

TECH ist die größte digitale Universität der Welt. Mit einem beeindruckenden Katalog von über 14.000 Hochschulprogrammen, die in 11 Sprachen angeboten werden, ist sie mit einer Vermittlungsquote von 99% führend im Bereich der Beschäftigungsfähigkeit. Darüber hinaus verfügt sie über einen beeindruckenden Lehrkörper mit mehr als 6.000 Professoren von höchstem internationalem Prestige.

Studieren Sie an der größten digitalen Universität der Welt und sichern Sie sich Ihren beruflichen Erfolg. Die Zukunft beginnt bei TECH“

Die beste Online-Universität der Welt laut FORBES

Das renommierte, auf Wirtschaft und Finanzen spezialisierte Magazin Forbes hat TECH als „beste Online-Universität der Welt" ausgezeichnet. Dies wurde kürzlich in einem Artikel in der digitalen Ausgabe des Magazins festgestellt, in dem die Erfolgsgeschichte dieser Einrichtung „dank ihres akademischen Angebots, der Auswahl ihrer Lehrkräfte und einer innovativen Lernmethode, die auf die Ausbildung der Fachkräfte der Zukunft abzielt", hervorgehoben wird.

Die besten internationalen Top-Lehrkräfte

Der Lehrkörper der TECH besteht aus mehr als 6.000 Professoren von höchstem internationalen Ansehen. Professoren, Forscher und Führungskräfte multinationaler Unternehmen, darunter Isaiah Covington, Leistungstrainer der Boston Celtics, Magda Romanska, leitende Forscherin am Harvard MetaLAB, Ignacio Wistumba, Vorsitzender der Abteilung für translationale Molekularpathologie am MD Anderson Cancer Center, und D.W. Pine, Kreativdirektor des TIME Magazine, um nur einige zu nennen.

Die größte digitale Universität der Welt

TECH ist die weltweit größte digitale Universität. Wir sind die größte Bildungseinrichtung mit dem besten und umfangreichsten digitalen Bildungskatalog, der zu 100% online ist und die meisten Wissensgebiete abdeckt. Wir bieten weltweit die größte Anzahl eigener Abschlüsse sowie offizieller Grund- und Aufbaustudiengänge an. Insgesamt sind wir mit mehr als 14.000 Hochschulabschlüssen in elf verschiedenen Sprachen die größte Bildungseinrichtung der Welt.

Die umfassendsten Lehrpläne in der Universitätslandschaft

TECH bietet die vollständigsten Lehrpläne in der Universitätslandschaft an, mit Lehrplänen, die grundlegende Konzepte und gleichzeitig die wichtigsten wissenschaftlichen Fortschritte in ihren spezifischen wissenschaftlichen Bereichen abdecken. Darüber hinaus werden diese Programme ständig aktualisiert, um den Studenten die akademische Avantgarde und die gefragtesten beruflichen Kompetenzen zu garantieren. Auf diese Weise verschaffen die Abschlüsse der Universität ihren Absolventen einen bedeutenden Vorteil, um ihre Karriere erfolgreich voranzutreiben.

Eine einzigartige Lernmethode

TECH ist die erste Universität, die *Relearning* in allen ihren Studiengängen einsetzt. Es handelt sich um die beste Online-Lernmethodik, die mit internationalen Qualitätszertifikaten renommierter Bildungseinrichtungen ausgezeichnet wurde. Darüber hinaus wird dieses disruptive akademische Modell durch die „Fallmethode" ergänzt, wodurch eine einzigartige Online-Lehrstrategie entsteht. Es werden auch innovative Lehrmittel eingesetzt, darunter ausführliche Videos, Infografiken und interaktive Zusammenfassungen.

Die offizielle Online-Universität der NBA

TECH ist die offizielle Online-Universität der NBA. Durch eine Vereinbarung mit der größten Basketball-Liga bietet sie ihren Studenten exklusive Universitätsprogramme sowie eine breite Palette von Bildungsressourcen, die sich auf das Geschäft der Liga und andere Bereiche der Sportindustrie konzentrieren. Jedes Programm hat einen einzigartig gestalteten Lehrplan und bietet außergewöhnliche Gastredner: Fachleute mit herausragendem Sporthintergrund, die ihr Fachwissen zu den wichtigsten Themen zur Verfügung stellen.

Führend in Beschäftigungsfähigkeit

TECH ist es gelungen, die führende Universität im Bereich der Beschäftigungsfähigkeit zu werden. 99% der Studenten finden innerhalb eines Jahres nach Abschluss eines Studiengangs der Universität einen Arbeitsplatz in dem von ihnen studierten Fachgebiet. Ähnlich viele erreichen einen unmittelbaren Karriereaufstieg. All dies ist einer Studienmethodik zu verdanken, die ihre Wirksamkeit auf den Erwerb praktischer Fähigkeiten stützt, die für die berufliche Entwicklung absolut notwendig sind.

Google Partner Premier

Der amerikanische Technologieriese hat TECH mit dem Logo Google Partner Premier ausgezeichnet. Diese Auszeichnung, die nur 3% der Unternehmen weltweit erhalten, unterstreicht die effiziente, flexible und angepasste Erfahrung, die diese Universität den Studenten bietet. Die Anerkennung bestätigt nicht nur die maximale Präzision, Leistung und Investition in die digitalen Infrastrukturen der TECH, sondern positioniert diese Universität auch als eines der modernsten Technologieunternehmen der Welt.

Die von ihren Studenten am besten bewertete Universität

Die Studenten haben TECH auf den wichtigsten Bewertungsportalen als die am besten bewertete Universität der Welt eingestuft, mit einer Höchstbewertung von 4,9 von 5 Punkten, die aus mehr als 1.000 Bewertungen hervorgeht. Diese Ergebnisse festigen die Position der TECH als internationale Referenzuniversität und spiegeln die Exzellenz und die positiven Auswirkungen ihres Bildungsmodells wider.

03

Lehrplan

Im Laufe des Programms werden die Fachleute die Grundlagen von Algorithmen, Datenstrukturen und Programmierlogik behandeln und sich dann mit Sprachen wie *Python*, *JavaScript* und *HTML* befassen. Darüber hinaus werden sie sich mit wesentlichen Bereichen wie Webdesign, Anwendungsentwicklung und Datenbankmanagement befassen und einen umfassenden Überblick über die Welt der Programmierung geben. Dank eines progressiven und strukturierten Ansatzes gewährleistet dieser Lehrplan nicht nur den Erwerb von technischem Wissen, sondern auch die Entwicklung analytischer und kreativer Fähigkeiten, die in einem wettbewerbsorientierten Arbeitsumfeld unerlässlich sind.

Dieser Aufbaustudiengang ist die ideale Wahl, um Ihre Karriere im Technologiesektor zu beginnen und sich die notwendigen Werkzeuge anzueignen, um sich in jedem Berufsfeld der Programmierung zu profilieren“

Modul 1. Programmierung und Softwareentwicklung von Grund auf

1.1. Software und Hardware. Beziehung und Unterschiede
- 1.1.1. Software
- 1.1.2. Unterschiede zwischen Software und Hardware
- 1.1.3. Beziehung zwischen Software, Hardware und Programmierung

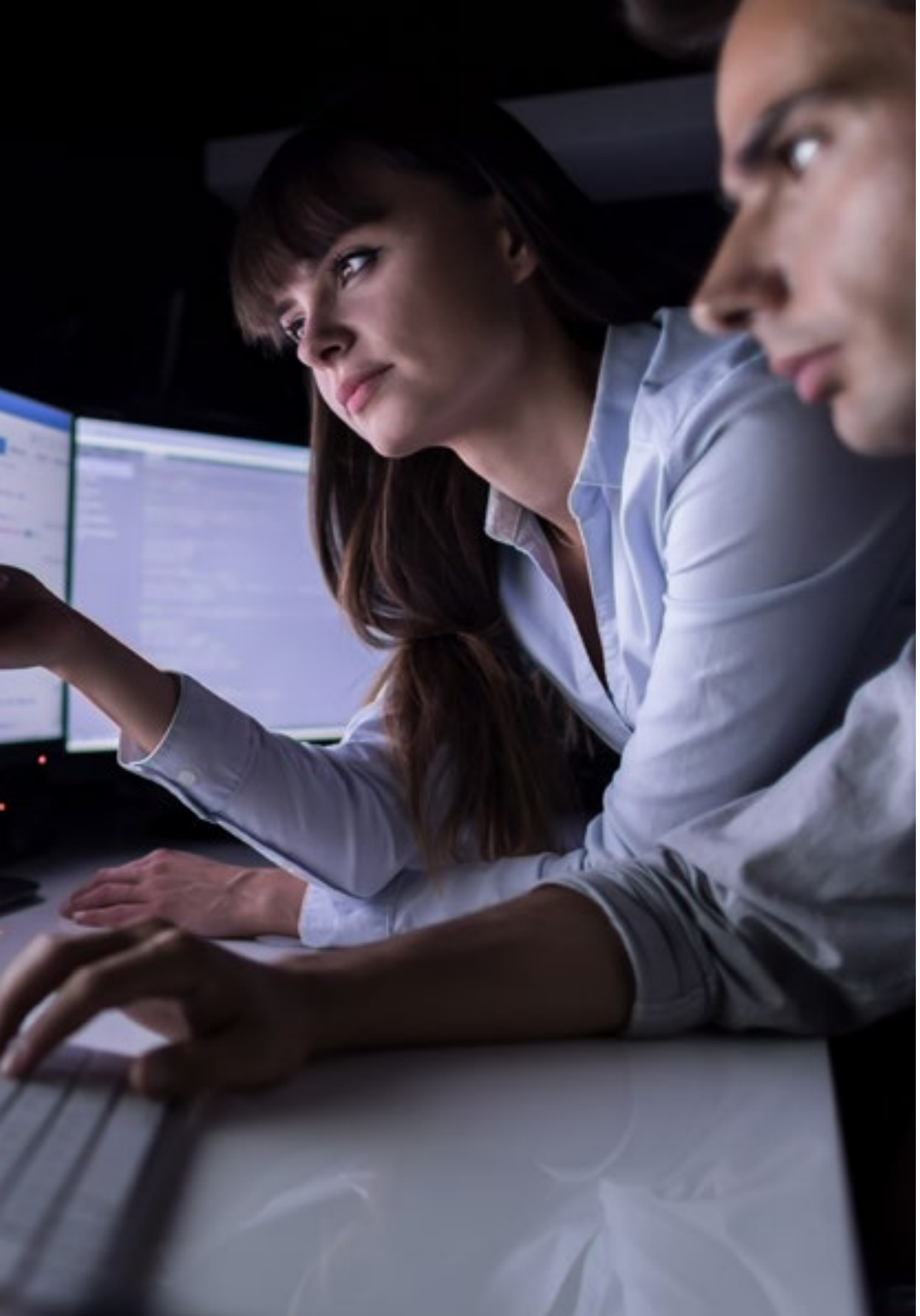
1.2. Programmierung. Schlüsselaspekte
- 1.2.1. Programmierung
- 1.2.2. Ziele und Anwendungen
- 1.2.3. Programme, Quellcode, Kompilierung und Ausführung
- 1.2.4. Fehler: Syntaxfehler, Ausführungsfehler, logische Fehler

1.3. Programmierung von Grund auf
- 1.3.1. Struktur eines Programms
- 1.3.2. Algorithmen: Struktur und Beispiele
- 1.3.3. Beziehung zwischen Algorithmen und Programmen
- 1.3.4. Lösen von Problemen durch Algorithmen

1.4. Paradigmen und Arten von Programmiersprachen
- 1.4.1. Paradigmen der Programmierung
  - 1.4.1.1. Imperatives Paradigma
  - 1.4.1.2. Objektorientiertes Paradigma
  - 1.4.1.3. Funktionales Paradigma
  - 1.4.1.4. Deklaratives Paradigma
- 1.4.2. Low-Level- vs. High-Level-Sprachen
- 1.4.3. Kompilierte vs. interpretierte Sprachen

1.5. Übersetzung von Programmiersprachen
- 1.5.1. Der Compiler. Kompilierungsprozess
- 1.5.2. Der Interpreter: Interpretationsprozess
- 1.5.3. Unterschiede zwischen Kompilierung und Interpretation

1.6. Bits, binäre Operationen und logische Gatter
- 1.6.1. Bit. Binäre Darstellung
- 1.6.2. Grundlegende Bitoperationen: AND, OR, XOR, NOT
- 1.6.3. Umrechnung zwischen binärer und dezimaler Darstellung
- 1.6.4. Logikgatter: OR, AND, XOR, NOT, NOR y NAND



1.7. Entwurf von Algorithmen von Grund auf
  1.7.1. Entwurf eines Algorithmus von Grund auf
  1.7.2. Sequentielle, bedingte und repetitive Algorithmen
  1.7.3. Rekursive Algorithmen und ihr Vergleich mit iterativen Algorithmen
1.8. Elemente des Programms: Komponenten und Struktur
  1.8.1. Dateneingabe und -ausgabe
  1.8.2. Variablen und Konstanten: Nutzung
  1.8.3. Datenverarbeitung und -manipulation
  1.8.4. Grundlegende Funktionen und Verfahren mit Flussdiagrammen
1.9. Kontrollstrukturen mit Flussdiagrammen
  1.9.1. Kontrollstrukturen. Funktion in der Programmierung
  1.9.2. Bedingte Strukturen: Beispiele mit Flussdiagrammen
  1.9.3. Sich wiederholende Strukture: Beispiele mit Flussdiagrammen
1.10. Lebenszyklus und Softwareentwicklungsmodelle
  1.10.1. Lebenszyklus der Software. Phasen
  1.10.2. Entwicklungsmodelle: Wasserfall, iterativ und agil
  1.10.3. Testing und Wartung bei der Softwareentwicklung

Modul 2. Backend-Entwicklung I: Python von Grund auf

2.1. Python von Grund auf. Installation
  2.1.1. Programmiersprache Python. Merkmale
  2.1.2. Installation von Python unter Windows, macOS und Linux
  2.1.3. Konfiguration der Entwicklungsumgebung: IDEs und Code-Editoren
  2.1.4. Erstes Programm in Python: „Hallo Welt“
2.2. Syntax und Variablen in Python
  2.2.1. Struktur des Codes in Python: Einrückung
  2.2.2. Kommentare in Python
  2.2.3. Variablen und Datentypen in Python
  2.2.4. Arithmetische und logische Operationen in Python
2.3. Flusskontrolle: Bedingungen
  2.3.1. Kontrollstrukturen
  2.3.2. Bedingungen: if, elif, else
  2.3.3. Ternäre Bedingung

2.4. Schleifen in Python
- 2.4.1. Verwendung von Schleifen in der Programmierung
- 2.4.2. „for"- und „while"-Schleifen
- 2.4.3. Ablaufsteuerung in Schleifen: break und continue
- 2.4.4. Verschachtelte Schleifen

2.5. Funktionen in Python
- 2.5.1. Funktion in Python. Nutzen
- 2.5.2. Parameter und Argumente der Funktionen
- 2.5.3. Rückgabewerte
- 2.5.4. Vordefinierte Funktionen vs. benutzerdefinierte Funktionen

2.6. Listen und Tupel in Python
- 2.6.1. Erstellung und Verwendung von Listen in Python
- 2.6.2. Allgemeine Operationen mit Listen: Hinzufügen, Löschen, Ändern
- 2.6.3. Tupel: Unterschiede zu Listen
- 2.6.4. Iteration über Listen und Tupel

2.7. Wörterbücher und Sätze in Python
- 2.7.1. Wörterbücher: Schlüssel-Wert
- 2.7.2. Methoden zur Bearbeitung von Wörterbüchern
- 2.7.3. Sets: Nutzung
- 2.7.4. Vergleich von Wörterbüchern und Sets

2.8. Verwaltung von Dateien in Python von Grund auf
- 2.8.1. Öffnen und Schließen von Dateien
- 2.8.2. Öffnungsmodi: Lesen, Schreiben und Hinzufügen
- 2.8.3. Lesen und Schreiben von Textdateien

2.9. Umgang mit Fehlern und Ausnahmen
- 2.9.1. Arten von Ausnahmen
- 2.9.2. Verwendung von Try, Except zur Fehlerbehandlung
- 2.9.3. Erstellen von benutzerdefinierten Ausnahmen

2.10. Bewährte Verfahren und Debugging in Python
- 2.10.1. Debugging: Zweck
- 2.10.2. Debugging-Techniken: Verwendung von print und breakpoints
- 2.10.3. Bewährte Verfahren beim Schreiben von Code

Modul 3. Backend-Entwicklung II – Algorithmen und Datenstrukturen mit Python von Grund auf

3.1. Suchalgorithmen in Datenstrukturen
- 3.1.1. Zweck von Suchalgorithmen in Datenstrukturen
- 3.1.2. Lineare Suche: Implementierung und Anwendungsfälle
- 3.1.3. Binäre Suche: Beispiele
- 3.1.4. Vergleich der Effizienz: Lineare vs. binäre Suche

3.2. Sortieralgorithmen in Datenstrukturen (I). Grundlegende Sortiertechniken Bubble Sort und Insertion Sort
- 3.2.1. Blasen-Sortierung (Bubble Sort): Implementierung und Analyse
- 3.2.2. Einfügesortieren (Insertion Sort): Implementierung und Anwendungsfälle
- 3.2.3. Vergleich zwischen Bubble Sort und Insertion Sort

3.3. Sortieralgorithmen in Datenstrukturen (II). Fortgeschrittene Sortiertechniken Selection Sort, Merge Sort und Quick Sort
- 3.3.1. Selection Sort. Implementierung und Analyse
- 3.3.2. Merge Sort: Implementierung
- 3.3.3. Quick Sort: Implementierung
- 3.3.4. Effizienzvergleich zwischen Sortieralgorithmen

3.4. Rekursive Suchalgorithmen in Datenstrukturen
- 3.4.1. Rekursion. Nutzung
- 3.4.2. Direkte und indirekte Rekursion
- 3.4.3. Rekursive Algorithmen: Faktoriell und Fibonacci. Beispiele

3.5. Komplexität von Suchalgorithmen in Datenstrukturen
- 3.5.1. Algorithmische Komplexität. Messung der Effizienz
- 3.5.2. Big-O-Notation
- 3.5.3. Analyse der Komplexität in Such- und Sortieralgorithmen

3.6. Fortgeschrittene Datenstrukturen
- 3.6.1. Bäume: Terminologie
- 3.6.2. Binäre Bäume: Operationen
- 3.6.3. Binäre Suchbäume (BST): Suchen, Einfügen und Löschen

3.7. Graphalgorithmen
- 3.7.1. Graphen. Darstellung
- 3.7.2. Graphalgorithmen: DFS und BFS
- 3.7.3. Vergleich zwischen DFS und BFS

3.8. Dynamische Programmierung
- 3.8.1. Dynamische Programmierung. Anwendung
- 3.8.2. Unterschiede zwischen dynamischer Programmierung und Rekursion
- 3.8.3. Optimierung durch dynamische Programmierung

3.9. Optimierungstechniken für Daten-Suchalgorithmen
- 3.9.1. Die Bedeutung der algorithmischen Optimierung der Datensuche
- 3.9.2. Optimierungstechniken: Memoisierung (Memoization)
- 3.9.3. *Divide and Conquer*: Aufteilen und Erobern

3.10. Andere Python-Algorithmen
- 3.10.1. Permutations- und Kombinationsalgorithmen
- 3.10.2. Grundlegende Hash-Algorithmen
- 3.10.3. Zählalgorithmen und Algorithmen zur Erzeugung von Teilmengen

Modul 4. Backend-Entwicklung III – Flask, Erstellung von APIs und grundlegende Architektur von Grund auf

4.1. Flask als Backend-Framework
- 4.1.1. Backend-Framework. Zweck
- 4.1.2. Flask. Merkmale
- 4.1.3. Vorbereitung der Entwicklungsumgebung und Installation von Flask
- 4.1.4. Erstes Projekt mit Flask: „Hallo Welt“

4.2. Routen und HTTP-Anfragen in Flask
- 4.2.1. Route: Funktionsweise in einer Webanwendung
- 4.2.2. HTTP-Methoden in Flask: GET, POST, PUT und DELETE
- 4.2.3. Routen in Flask mit Parametern und Daten
- 4.2.4. Organisation von Routen in einem Projekt

4.3. Controller und Umgang mit Antworten in Flask
- 4.3.1. Controller: Rollen und Verantwortlichkeiten
- 4.3.2. Arten von Antworten in Flask: Text, JSON und HTML
- 4.3.3. Erstellung von Controllern für APIs in Flask
- 4.3.4. CRUD-Operationen in Controllern

4.4. RESTful-APIs
- 4.4.1. RESTful-API. Grundsätze
- 4.4.2. HTTP-Methoden und Konventionen in RESTful-APIs
- 4.4.3. Erstellung einer RESTful-API in Flask
- 4.4.4. Entwurf einer Aufgaben-API mit CRUD-Operationen

4.5. Datenbanken und Flask mit SQLite
- 4.5.1. Datenbanken in Webanwendungen
- 4.5.2. Verbindung zu SQLite in Flask-Projekten
- 4.5.3. Erstellung von Tabellen und Modellen mit SQLAlchemy
- 4.5.4. CRUD-Abfragen in SQLite zur Datenverwaltung

4.6. Authentifizierung und grundlegende Sicherheit in APIs
- 4.6.1. Authentifizierung und Autorisierung in APIs
- 4.6.2. Erstellung eines Benutzerauthentifizierungssystems
- 4.6.3. Verwendung von Token zur Authentifizierung in Flask
- 4.6.4. Schutz von Routen und Benutzerdaten in APIs

4.7. Datenvalidierung und Fehlerbehandlung
- 4.7.1. Fehlerbehandlung und Ausnahmen in Flask
- 4.7.2. Datenvalidierung bei API-Anfragen
- 4.7.3. Erstellung personalisierter Fehlermeldungen
- 4.7.4. Validierungsstrategien und Fehlerbehandlung bei CRUD

4.8. Strukturierung skalierbarer APIs
- 4.8.1. Organisation und Struktur eines skalierbaren Flask-Projekts
- 4.8.2. Modularisierung und Aufteilung der Zuständigkeiten bei APIs
- 4.8.3. Grundlegende Optimierung von APIs für Leistung und Skalierbarkeit
- 4.8.4. Organisationsstrategie für große Projekte

4.9. Echtzeitkommunikation mit WebSockets
- 4.9.1. WebSockets. Anwendungen
- 4.9.2. Implementierung von WebSockets in Flask mit Flask-SocketIO
- 4.9.3. Echtzeitkommunikation in Flask-Anwendungen

4.10. Deployment und Wartung von Anwendungen
- 4.10.1. Vorbereitung von Flask-Anwendungen für die Produktion
- 4.10.2. Deployment auf gängigen Plattformen wie Heroku und Render
- 4.10.3. Verwendung von Docker für das Deployment in Containern
- 4.10.4. Überwachung und Wartung von Backend-Anwendungen

Modul 5. Objektorientierte Programmierung und Entwurfsmuster von Grund auf

5.1. Objektorientierte Programmierung (OOP) von Grund auf
- 5.1.1. Objektorientiertes Programmieren
- 5.1.2. Unterschiede zwischen OOP und strukturierter Programmierung
- 5.1.3. OOP-Elemente: Klassen, Objekte, Methoden und Attribute

5.2. Klassen und Objekte in Python
- 5.2.1. Erstellung von Klassen und Objekten in Python
- 5.2.2. Instanz- und Klassenattribute
- 5.2.3. Spezielle Methoden (init, str, repr usw.)
- 5.2.4. Statische Methoden und Klassenmethoden: Verwendungen

5.3. Kapselung und Abstraktion in Klassen
- 5.3.1. Kapselung: Nutzen
- 5.3.2. Zugriffsmodifizierer in Python
  - 5.3.2.1. Öffentlich, geschützt und privat
- 5.3.3. Abstraktion: Verbergen von Details und Verbesserung der Einfachheit
- 5.3.4. Verwendung von Eigenschaften (@property) zur Zugriffskontrolle

5.4. Vererbung in Python. Nutzen in der OOP
- 5.4.1. Vererbung: Nutzen in der OOP
- 5.4.2. Erstellung abgeleiteter Klassen und Mehrfachvererbung in Python
- 5.4.3. Vererbte Methoden und Attribute und Überladung in der Vererbung
- 5.4.4. Klassenhierarchien und Verwaltung von Basisklassen

5.5. Polymorphismus und Überladung in Python
- 5.5.1. Polymorphismus: Duck Typing
- 5.5.2. Polymorphismus mit Klassen und Methoden in Python
- 5.5.3. Überladung und Überschreibung von Methoden in Python
- 5.5.4. Polymorphismus im Software-Design. Anwendungen und Vorteile

5.6. Beziehungen zwischen Klassen und Entwurf komplexer Strukturen
- 5.6.1. Arten von Beziehungen: Assoziation, Aggregation und Komposition
- 5.6.2. Unterschiede zwischen Aggregation und Komposition: Beispiele
- 5.6.3. Entwurf komplexer Strukturen unter Verwendung von Beziehungen zwischen Klassen

5.7. Entwurfsmuster und SOLID-Prinzipien
- 5.7.1. Bedeutung der Entwurfsmuster
- 5.7.2. Anwendung von Entwurfsmustern in OOP-Projekten. Vorteile
- 5.7.3. Klassifizierung von Entwurfsmustern
- 5.7.4. SOLID-Prinzipien und ihre Bedeutung für den objektorientierten Entwurf

5.8. Erzeugungsmuster
- 5.8.1. Zweck der Erzeugungsmuster
- 5.8.2. *Singleton*-Muster
- 5.8.3. *Factory*-Muster und *Factory Method*
- 5.8.4. *Builder*-Muster

5.9. Strukturmuster
- 5.9.1. Zweck der Strukturmuster
- 5.9.2. *Adapter*-Muster
- 5.9.3. *Decorator*-Muster
- 5.9.4. *Facade*-Muster

5.10. Verhaltensmuster
- 5.10.1. Verhaltensmuster. Anwendungen
- 5.10.2. *Observer*-Muster

5.10.3. *Strategy*-Muster

Modul 6. Frontend I – HTML und CSS von Grund auf

6.1. HTML von Grund auf
- 6.1.1. HTML. Zweck in der Webentwicklung
- 6.1.2. Struktur eines HTML-Dokuments: DOCTYPE, <html>, <head>, <body>
- 6.1.3. Semantische und Inhalts-Tags: <header>, <nav>, <section>, <footer>
- 6.1.4. Wesentliche Elemente: Absätze (<p>), Listen (<ul>, <ol>), Links (<a>), Bilder (<img>)
- 6.1.5. Bewährte Verfahren in HTML

6.2. Text- und Multimediaelemente in HTML
- 6.2.1. Grundlegende Text-Tags: <p>, Überschriften, Listen, Fett- und Kursivschrift
- 6.2.2. Einfügen von Multimedia: Attribute von <img>, <audio>, <video>
- 6.2.3. Grundlegende Attribute für die Barrierefreiheit (alt, aria-label)

6.3. HTML-Formulare
- 6.3.1. Struktur und Komponenten von Formularen: <form>, <input>, <label>, <button>
- 6.3.2. Eingabetypen: Text, E-Mail, Passwort, Senden-Schaltflächen
- 6.3.3. HTML5-Validierung: Clientseitige Feldvalidierung
- 6.3.4. Formulare mit einfacher Validierung. Beispiele

6.4. CSS von Grund auf
- 6.4.1. CSS-Sprache von Grund auf: Verwendung und Beziehung zu HTML
- 6.4.2. CSS-Syntax: Selektoren, Eigenschaften und Werte
- 6.4.3. Anwendung von Inline-, Internal- und External-Styles
- 6.4.4. Erweiterte Selektoren: Typ, Klasse, ID, Pseudoklassen (:hover, :focus)

6.5. Box-Modell in CSS
- 6.5.1. Box-Modell: Bedeutung in CSS
- 6.5.2. Schlüsselmerkmale: margin, padding, border, width, height
- 6.5.3. Verwendung von box-sizing zur präzisen Steuerung des Box-Modells
- 6.5.4. Das auf das Box-Modell angewandte Design. Beispiele

6.6. Typografie und Textstile in CSS
- 6.6.1. Eigenschaften von Farben und Schriftarten: Farbe, font-family, font-size
- 6.6.2. Erweiterte Textstile: Fettdruck, Kursivschrift, Ausrichtung (text-align)
- 6.6.3. Abstand und Trennung von Text: line-height, letter-spacing
- 6.6.4. Maßeinheiten in CSS (px, em, rem) und ihre Verwendung in der Typografie

6.7. Layoutgestaltung mit CSS - Flexbox
- 6.7.1. Flexbox: Zweck
- 6.7.2. Flexbox-Eigenschaften: justify-content, align-items, flex-direction
- 6.7.3. Verteilung von Elementen und Ausrichtung in Flexbox
- 6.7.4. Beispiele für Layouts mit Flexbox

6.8. CSS Grid und *Responsive Design* mit CSS
- 6.8.1. CSS Grid: Zeilen, Spalten und Bereiche
- 6.8.2. Media Queries: Struktur und Anwendung auf verschiedenen Geräten
- 6.8.3. *Responsive Design* für Mobilgeräte, Tablets und Desktops
- 6.8.4. Anpassung von Typografie und fließenden Einheiten im *Responsive Design*

6.9. Animationen und Übergänge in CSS
- 6.9.1. Übergänge: Eigenschaft transition, Effekte bei :hover
- 6.9.2. Animationen mit CSS: Verwendung von @keyframes, grundlegende Animationen
- 6.9.3. Techniken zur Glättung von Übergängen und Animationen im Web

6.10. Barrierefreiheit im Webdesign
- 6.10.1. Barrierefreiheit im Internet: Bedeutung
- 6.10.2. Gestaltung barrierefreier Websites. Bewährte Verfahren
- 6.10.3. Attribut aria-label und Tools zur Validierung der Barrierefreiheit

Modul 7. Frontend II – JavaScript von Grund auf

7.1. JavaScript von Grund auf
- 7.1.1. JavaScript Sprache
- 7.1.2. Integration von JavaScript in HTML
- 7.1.3. Erstes Programm in JavaScript: „Hallo Welt"

7.2. Variablen und Datentypen in JavaScript
- 7.2.1. Deklaration von Variablen mit var, let und const
- 7.2.2. Datentypen: Zahlen, Ketten, Boolesche Werte
- 7.2.3. Konvertierung zwischen Datentypen

7.3. Kontrollstrukturen in JavaScript
- 7.3.1. Bedingungen: if, else if, else
- 7.3.2. Schleifen: for, while, do...while
- 7.3.3. Switch-case: Alternative zu mehreren Bedingungen
- 7.3.4. Break und continue in Schleifen

7.4. Funktionen in JavaScript
- 7.4.1. Funktionsdeklarationen
- 7.4.2. Parameter, Rückgabewerte und Gültigkeitsbereich
- 7.4.3. Pfeilfunktionen (=>) und anonyme Funktionen
- 7.4.4. Callbacks und Rekursion in Funktionen

7.5. DOM-Manipulation (Document Object Model) mit JavaScript
- 7.5.1. DOM: Struktur eines HTML-Dokuments
- 7.5.2. Auswahl von DOM-Elementen (getElementById, querySelector)
- 7.5.3. Manipulation von Elementen: Ändern von Text, Stilen und Attributen
- 7.5.4. Ereignisse: Klick, Eingabe, Absenden und mehr

7.6. Arrays und Objekte in JavaScript
- 7.6.1. Deklaration und Verwendung von Arrays
- 7.6.2. Gängige Array-Methoden: push, pop, map, filter
- 7.6.3. Erstellung und Verwendung von Objekten
- 7.6.4. Iteration über Arrays und Objekte

7.7. Promises und Asynchronität in JavaScript
- 7.7.1. Asynchronität und Verwendung von Callbacks in JavaScript
- 7.7.2. Promise in JavaScript: Erstellung und Handhabung
- 7.7.3. Verwendung von async und await in JavaScript

7.8. APIs und Fetch in JavaScript
- 7.8.1. API in JavaScript: Zweck
- 7.8.2. Verwendung von REST-APIs mit fetch
- 7.8.3. Fehlerbehandlung und Status von Anfragen

7.9. Lokaler Speicher in Webbrowsern
- 7.9.1. Local Storage und Session Storage in Diensten und Webanwendungen
- 7.9.2. Speichern und Abrufen von Daten im Local Storage
- 7.9.2. IndexedDB als Datenbank für den Browser
- 7.9.3. Verwaltung von Cookies in JavaScript
- 7.9.4. Speicherung im Browser: Beispiele

7.10. Bewährte Verfahren in JavaScript und Entwicklertools
- 7.10.1. JavaScript-Code: Bewährte Verfahren
- 7.10.2. Verwendung von Browser-Entwicklertools in JavaScript
- 7.10.3. Debugging und Fehlerbehandlung in JavaScript

Modul 8. Frontend III – React.js von Grund auf

8.1. React.js von Grund auf
- 8.1.1. React JS als Bibliothek für die Entwicklung von Webanwendungen
- 8.1.2. Komponenten und Virtual DOM in React JS: Architektur und Funktionsweise
- 8.1.3. Installation und Konfiguration mit dem NextJS-Framework
- 8.1.4. Erste Komponente in React: „Hallo Welt"

8.2. JavaScript XML oder JSX und Komponenten in React
- 8.2.1. JSX: Syntax und Merkmale
- 8.2.2. Erstellung funktionaler Komponenten in React.js
- 8.2.3. Verwendung von props für die Datenübertragung zwischen Komponenten
- 8.2.4. Funktionale Komponenten im Vergleich zu Klassenkomponenten für die Entwicklung in React.js

8.3. Status und Ereignisse in React.js
- 8.3.1. Status der Komponenten in React.js
- 8.3.2. Verwendung von useState für die Statusverwaltung
- 8.3.3. Verwaltung von Ereignissen in React.js: onClick, onChange u. a.
- 8.3.4. Beispiele für die Status- und Ereignisverwaltung in React.js

8.4. Lebenszyklus von Komponenten und Effekten in React
- 8.4.1. Lebenszyklus von Komponenten in React
- 8.4.2. Verwendung von useEffect zur Verwaltung von Effekten in React
- 8.4.3. Komponenten mit Montage, Aktualisierung und Demontage in React

8.5. Routing mit React Router
- 8.5.1. SPA (Single Page Applications) und Routing in Webanwendungen
- 8.5.2. Installation und Konfiguration von React Router
- 8.5.3. Erstellung von Routen und Navigation zwischen Seiten mit React Router

8.6. Formulare und Validierung in React
- 8.6.1. Erstellung interaktiver Formulare in React
- 8.6.2. Verarbeitung von Benutzereingaben und Senden von Daten in React
- 8.6.3. Validierung von Formularen in Echtzeit in React

8.7. Verwendung von APIs in React
- 8.7.1. Verwendung von APIs mit fetch und axios in React
- 8.7.2. Verarbeitung von Lade-, Erfolgs- und Fehlerstatus in React
- 8.7.3. Aktualisierung von Komponenten gemäß API-Daten in React

8.8. Wiederverwendbare Komponenten und externe Bibliotheken in React
- 8.8.1. Wiederverwendbare Komponenten in React
- 8.8.2. Erstellung wiederverwendbarer Komponenten in React
- 8.8.3. Verwendung externer Bibliotheken wie Material UI und Bootstrap in React

8.9. Globale Zustandsverwaltung in React
- 8.9.1. Globale Zustandsverwaltung mit nativen Optionen: Context API und benutzerdefinierte Hooks
- 8.9.2. Externe Bibliotheken für die Datenverwaltung
- 8.9.3. Vergleich von Ansätzen für die globale Zustandsverwaltung. Beispiele

8.10. Deployment und Optimierung von React-Anwendungen
- 8.10.1. Vorbereitung einer React-Anwendung für die Produktion
- 8.10.2. Deployment auf Plattformen wie Netlify und Vercel
- 8.10.3. Leistungsoptimierung: Lazy Loading, Memoization, Serverkomponenten und Code Splitting
- 8.10.4. Überwachung und Wartung von React-Anwendungen in der Produktion. Tools und Leistungsanalyse

Modul 9. Verwaltung und Optimierung von Datenbanken von Grund auf

9.1. Datenbank von Grund auf
- 9.1.1. Datenbanken: Arten
- 9.1.2. Relationale vs. nicht relationale Datenbanken
- 9.1.3. SQL- und NoSQL-Programmiersprachen

9.2. Modellierung relationaler Daten
- 9.2.1. Relationales Datenbankmodell
- 9.2.2. Tabellen, Zeilen und Spalten in einer relationalen Datenbank
- 9.2.3. Primär- und Fremdschlüssel: Beziehungen zwischen Tabellen
- 9.2.4. Normalisierung: 1NF, 2NF, 3NF

9.3. SQL-Sprache: DML und DDL
- 9.3.1. SQL: Strukturierte Abfragesprache
- 9.3.2. Erstellungs- und Löschabfragen: CREATE, DROP
- 9.3.3. SELECT-, INSERT-, UPDATE- und DELETE-Abfrage
- 9.3.4. Filtern und Sortieren von Daten mit SQL

9.4. Erweiterte Abfragen in SQL
- 9.4.1. Joins: INNER JOIN und OUTER JOIN
- 9.4.2. Unterabfragen und verschachtelte Abfragen
- 9.4.3. Aggregierte Funktionen in SQL: SUM, AVG, COUNT

9.5. NoSQL-Datenbanken und MongoDB
- 9.5.1. NoSQL-Datenbank
- 9.5.2. Vergleich zwischen SQL und NoSQL
- 9.5.3. MongoDB: Dokumentendatenbank
- 9.5.4. Flexible Schemata in NoSQL

9.6. Datenbankoptimierung
- 9.6.1. Bedeutung der Abfrageoptimierung
- 9.6.2. Verwendung von Indizes in relationalen Datenbanken
- 9.6.3. Optimierung von NoSQL-Datenbanken

9.7. Sicherheit in Datenbanken
- 9.7.1. Sicherheit in Datenbanken
- 9.7.2. Verschlüsselung sensibler Daten
- 9.7.3. Benutzer- und Berechtigungsverwaltung in Datenbanken
- 9.7.4. Strategien zum Schutz von Datenbanken vor Angriffen

9.8. Skalierbarkeit von Datenbanken
- 9.8.1. Skalierbarkeit in Datenbanken
- 9.8.2. Horizontale und vertikale Partitionierung
- 9.8.3. Replikation und Clustering in Datenbanken

9.9. Datensicherung und -wiederherstellung
- 9.9.1. Bedeutung der Datensicherung in Datenbanken
- 9.9.2. Automatische und manuelle Sicherungstechniken
- 9.9.3. Datenwiederherstellung in relationalen Datenbanken und in NoSQL-Datenbanken

9.10. Implementierung von Datenbanken in Projekten
- 9.10.1. Entwurf von Datenbanken für ein reales Projekt
- 9.10.2. Integration von Datenbanken mit Backend-Anwendungen

Modul 10. Tools für die Entwicklung von Grund auf: Linux, Versionskontrolle, CI/CD, Docker und agile Methoden

10.1. Linux von Grund auf
- 10.1.1. Linux
- 10.1.2. Unterschiede zwischen Linux und anderen Betriebssystemen
- 10.1.3. Beliebte Linux-Distributionen für Entwickler
- 10.1.4. Konfiguration und Anpassung der Entwicklungsumgebung
- 10.1.5. Texteditoren unter Linux

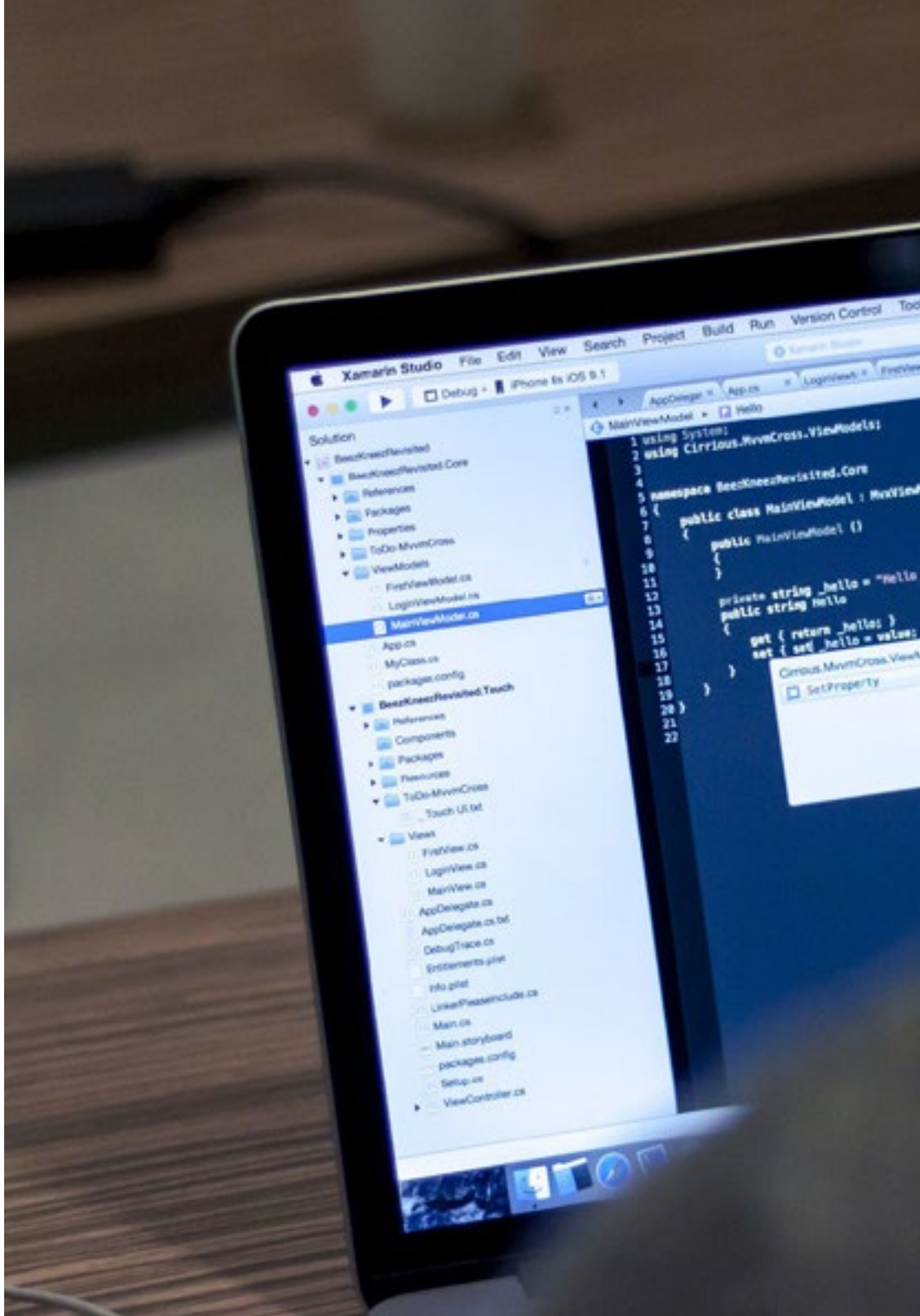
10.2. Verwendung des Terminals unter Linux von Grund auf
- 10.2.1. Das Terminal. Verwendung und Funktionen
- 10.2.2. Befehle zur Navigation und Dateiverwaltung im Terminal
- 10.2.3. Datei- und Verzeichnisberechtigungen unter Linux
- 10.2.4. Befehlsumleitung und Verwendung von Pipes zur Optimierung von Aufgaben

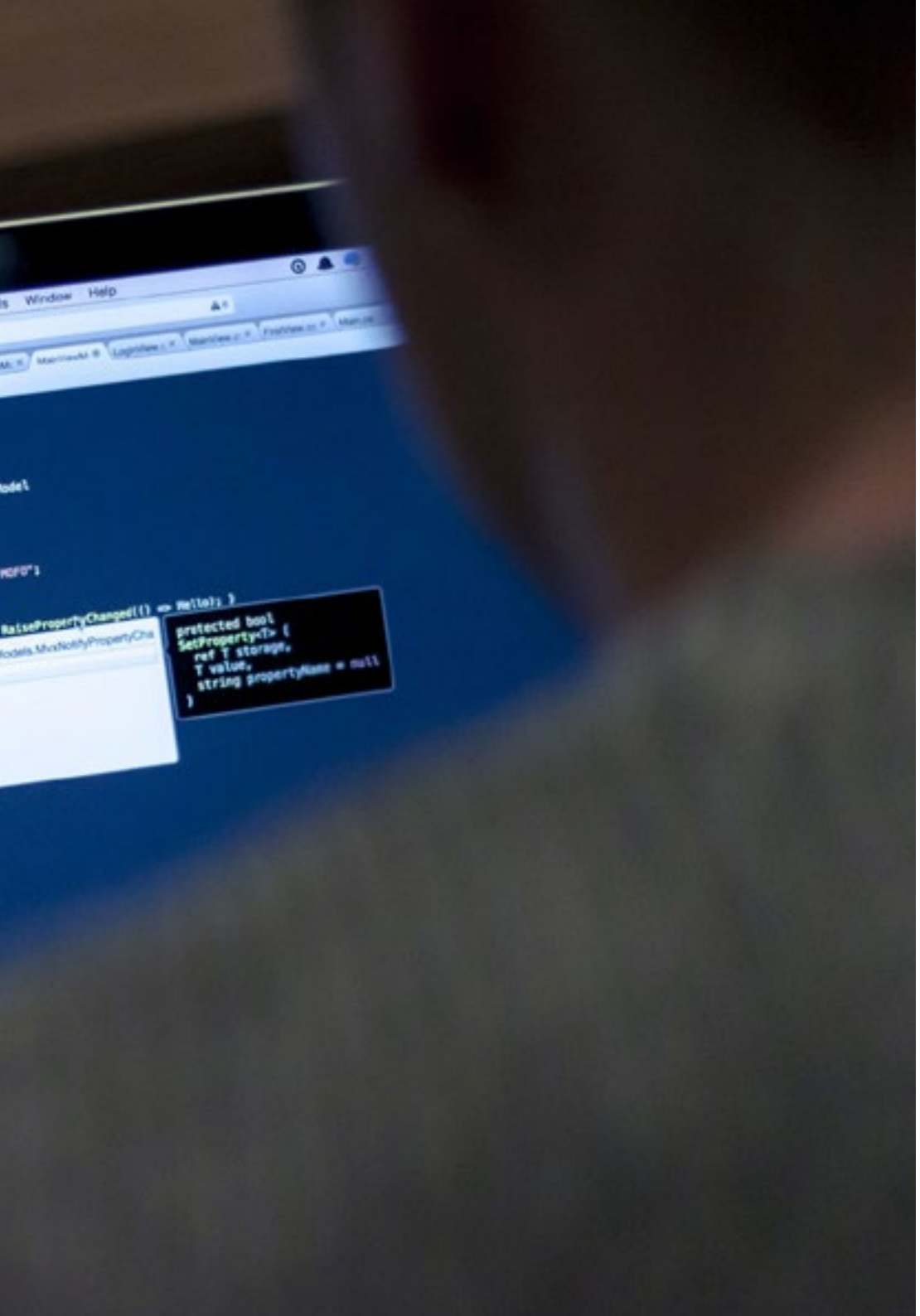
10.3. Versionskontrolle mit Git von Grund auf
- 10.3.1. Git: Cloud-Anbieter
- 10.3.2. Erstellung und Verwaltung von Repositorien
- 10.3.3. *Workflow*: git init, git add, git commit und git status
- 10.3.4. Arbeiten mit *Branches*: Erstellen, Zusammenführen und Auflösen von Konflikten

10.4. Zusammenarbeit in Teams mit GitHub von Grund auf
- 10.4.1. GitHub: *Remote-Repositories*
- 10.4.2. Verbindung eines lokalen Repositoriums mit GitHub: git remote. Erstkonfiguration
- 10.4.3. Synchronisation mit *Remote-Repositories*
- 10.4.4. Pull-Requests und kollaborative Code-Überprüfung

10.5. CI/CD (I) – Kontinuierliche Integration (CI) mit GitHub Actions von Grund auf
- 10.5.1. Die kontinuierliche Integration (CI)
- 10.5.2. Einrichten von *Workflows* in GitHub Actions
- 10.5.3. Automatisierung von Tests und Bereitstellungen

10.6. Docker von Grund auf
- 10.6.1. Docker und Container
- 10.6.2. Installation und Konfiguration von Docker
- 10.6.3. Erstellung und Verwaltung von Docker-Containern
- 10.6.4. Dockerfiles: Erstellung von benutzerdefinierten Bildern





10.7. CI/CD (II) – Kontinuierliche Bereitstellung (CD) mit Docker und GitHub Actions von Grund auf
  10.7.1. Kontinuierliche Bereitstellung (CD)
  10.7.2. Einrichtung von CD-Pipelines mit Docker und GitHub Actions
  10.7.3. Automatisierte Bereitstellung mit Docker Compose
10.8. Agile Methoden von Grund auf (I). Prinzipien und Werte
  10.8.1. Agile Methoden Grundsätze
  10.8.2. Das Agile Manifest: grundlegenden Werte und Prinzipien
  10.8.3. Vergleich mit traditionellen Methoden: Wasserfall vs. Agil
10.9. Agile Methoden (II): Scrum von Grund auf
  10.9.1. Scrum und seine Anwendbarkeit
  10.9.2. Schlüsselrollen in Scrum: Product Owner, Scrum Master und Development Team
  10.9.3. Artefakte in Scrum: Product Backlog, Sprint Backlog und Zuwachs
  10.9.4. Scrum-Events: Sprint Planning, Daily Scrum, Sprint Review und Retrospective
10.10. Agile Methoden (III): Kanban und Metriken von Grund auf
  10.10.1. Kanban und sein visueller Ansatz
  10.10.2. Schlüsselelemente in Kanban: Spalten, Karten und WIP-Limits
  10.10.3. Agile Metriken: Burnup, burndown charts, velocity und lead tim

Sie können sich bequem von zu Hause aus und in Ihrem eigenen Tempo mit der innovativen Online-Methodik und der Flexibilität, die Sie benötigen, fortbilden. Warten Sie nicht länger und bringen Sie Ihre Karriere in der Technologiebranche voran! Schreiben Sie sich noch heute ein“

04

Lehrziele

Dieser private Masterstudiengang zielt in erster Linie darauf ab, Fachleuten die notwendigen Fähigkeiten zu vermitteln, um ihre Karriere im Bereich der Programmierung solide und effektiv zu beginnen. Im Laufe des Programms werden sie ein tiefes Verständnis für die grundlegenden Konzepte der Programmierung entwickeln, wie Algorithmen, Datenstrukturen und Programmierlogik, die die Grundlage bilden, auf der sie fortgeschrittenere Fähigkeiten aufbauen werden. Darüber hinaus werden sie Programmiersprachen beherrschen, die in der Industrie weit verbreitet sind (*Python*, *JavaScript* und *HTML*), und so in der Lage sein, verschiedene technologische Projekte selbstbewusst in Angriff zu nehmen.

Sie werden die Fähigkeiten erwerben, die erforderlich sind, um Entwicklungsprojekte zu leiten, Datenbanken zu verwalten und Prozesse zu optimieren, und so Ihre berufliche Wettbewerbsfähigkeit im Technologiesektor steigern“

Allgemeine Ziele

- Beherrschen der Grundlagen der Softwareprogrammierung und -entwicklung durch Identifizierung der wesentlichen Elemente eines Programms, der Kontrollstrukturen und der Beziehung zwischen Software und Hardware
- Verstehen der Grundlagen von Python und seiner grundlegenden Syntax sowie Entwickeln einer geeigneten Umgebung für die Backend-Entwicklung
- Erwerben eines soliden Verständnisses grundlegender Algorithmen und Datenstrukturen
- Beherrschen der Grundlagen der Backend-Entwicklung und ihrer Rolle innerhalb einer Softwarearchitektur
- Verwalten der Prinzipien der objektorientierten Programmierung und deren Anwendung bei der Erstellung flexibler, wiederverwendbarer und wartbarer Software unter Verwendung von Python als Programmiersprache
- Entwickeln der grundlegenden Konzepte von HTML und CSS zur Strukturierung und Formatierung von Webseiten
- Analysieren der Sprache JavaScript, von den Grundlagen bis hin zu fortgeschrittenen Techniken
- Einführen der grundlegenden Konzepte von React.js zur Erstellung interaktiver Webanwendungen
- Verwalten von Datenbanktypen und ihrer Abfragesprachen, sowohl in relationalen als auch in NoSQL-Umgebungen
- Ausführliches Beschreiben der Automatisierung von Tests, der Integration und der Bereitstellung von Code

Spezifische Ziele

Modul 1. Programmierung und Softwareentwicklung von Grund auf

- Definieren und Unterscheiden von Software und Hardware
- Begründen der wesentlichen Konzepte der Programmierung
- Kennen der Grundstruktur eines Programms
- Erforschen und Analysieren der verschiedenen Programmierparadigmen

Modul 2. Backend-Entwicklung I: Python von Grund auf

- Beherrschen der Eigenschaften von Python
- Verstehen der grundlegenden Struktur und Syntax von Python
- Entwickeln von Fähigkeiten zur Ablaufsteuerung durch Bedingungen
- Anwenden von Schleifen zur Erstellung von Wiederholungszyklen in Python

Modul 3. Backend-Entwicklung II – Algorithmen und Datenstrukturen mit Python von Grund auf

- Implementieren und Vergleichen der verschiedenen Suchalgorithmen in Datenstrukturen
- Analysieren von Sortieralgorithmen wie Bubble, Insertion, Selection, Merge Sort und Quick Sort
- Untersuchen der algorithmischen Komplexität und Messung der Effizienz unter Verwendung der Big-O-Notation
- Darstellen von Graphen und Durchführen von Tiefensuche (DFS) und Breitensuche (BFS)

Modul 4. Backend-Entwicklung III – Flask, Erstellung von APIs und grundlegende Architektur von Grund auf

- Entwickeln von RESTful-APIs mit Flask
- Integrieren von Datenbanken in Flask-Anwendungen
- Implementieren von Authentifizierung und Sicherheit in APIs
- Entwerfen der grundlegenden Architektur einer Backend-Anwendung mit Flask

Modul 5. Objektorientierte Programmierung und Entwurfsmuster von Grund auf

- Definieren der Schlüsselkonzepte der objektorientierten Programmierung wie Klassen, Objekte, Attribute, Methoden, Kapselung, Abstraktion, Vererbung und Polymorphismus
- Verstehen der Verwendung von Kapselung und Abstraktion in Klassen mithilfe der Programmiersprache Python
- Untersuchen des Konzepts des Polymorphismus und der Überladung in der Sprache Python und Verstehen ihrer Anwendungen und Vorteile
- Bestimmen der Arten von Beziehungen zwischen Klassen wie Assoziation, Aggregation und Komposition

Modul 6. Frontend I – HTML und CSS von Grund auf

- Identifizieren der Grundstruktur eines HTML-Dokuments und ihrer Bedeutung für die Webentwicklung
- Verwenden von HTML zur semantischen und barrierefreien Organisation und Darstellung von Webinhalten: Web-Architektur
- Anwenden von CSS-Stilen zur Verbesserung der visuellen Darstellung von Elementen
- Verwenden des Box-Modells von CSS zur Strukturierung und Verteilung von Elementen in der Benutzeroberfläche

Modul 7. Frontend II – JavaScript von Grund auf

- Verstehen der Syntax und der Datentypen in JavaScript
- Erlernen der Strukturierung von Code mithilfe von Funktionen und Kontrollstrukturen
- Manipulieren des DOM zur Interaktion mit dynamischen Webseiten
- Arbeiten mit APIs und Handhaben von Asynchronität mithilfe von Promises und async/await

Modul 8. Frontend III – React.js von Grund auf

- Verstehen, wie JSX zur Erstellung deklarativer Schnittstellen verwendet wird
- Lernen, wie man mit funktionalen Komponenten, Props und Lebenszyklen arbeitet
- Verwalten von lokalen und globalen Zuständen mit modernen Tools wie Context API und Redux Toolkit
- Implementieren von Routing zur Erstellung von Single-Page-Anwendungen (SPAs)

Modul 9. Verwaltung und Optimierung von Datenbanken von Grund auf

- Erkennen der Arten von Datenbanken und ihrer Merkmale
- Verstehen und Anwenden des relationalen Datenmodells
- Entwickeln von SQL-Kenntnissen für die Datenbankverwaltung
- Verwenden fortgeschrittener Abfragen in SQL

Modul 10. Tools für die Entwicklung von Grund auf: Linux, Versionskontrolle, CI/CD, Docker und agile Methoden

- Bedienen des Betriebssystems Linux auf der Ebene der Befehlszeile
- Beherrschen der Verwendung von Git für die Versionskontrolle
- Implementieren von Pipelines für die kontinuierliche Integration und Bereitstellung (CI/CD)
- Erstellen und Verwalten von Docker-Containern

Möchten Sie die Sprache der Zukunft beherrschen? Dieser private Masterstudiengang ist der erste Schritt in die technologische Zukunft, die Sie sich schon immer gewünscht haben. Schreiben Sie sich jetzt ein"

05

Karrieremöglichkeiten

Dieser Abschluss eröffnet eine breite Palette von Karrieremöglichkeiten in einem sich ständig weiterentwickelnden Technologiesektor. Da Unternehmen in allen Branchen digitalisiert werden, steigt die Nachfrage nach Fachkräften mit Programmierkenntnissen nicht nur, sondern sie diversifiziert sich auch und eröffnet Möglichkeiten in so unterschiedlichen Bereichen wie Softwareentwicklung, Datenbankmanagement, Webdesign und Prozessautomatisierung. Nach Abschluss des Studiums sind die Absolventen bestens qualifiziert, um in Technologieunternehmen, innovativen Start-ups und Organisationen aller Branchen, die digitale Lösungen in ihre Abläufe integrieren möchten, Aufgaben zu übernehmen.

Sie werden nicht nur ein hochqualifizierter Experte, sondern auch eine Vielzahl von beruflichen Möglichkeiten nutzen können, um sich auf einem Arbeitsmarkt, der sich ständig weiterentwickelt und verändert, erfolgreich zu behaupten“

Profil des Absolventen

Der Absolvent wird ein hochqualifizierter Experte sein, der den technologischen Herausforderungen der Zukunft gewachsen ist. In diesem Sinne wird er nicht nur die grundlegenden Konzepte von Algorithmen, Datenstrukturen und Programmiersprachen beherrschen, sondern auch ein tiefes Verständnis für die in der Industrie am meisten nachgefragten Werkzeuge und Technologien entwickelt haben. Mit einer soliden Grundlage in Programmierung und Anwendungsentwicklung wird dieser Experte in der Lage sein, technologische Projekte selbstständig und effektiv anzugehen. Darüber hinaus wird er sich durch seine Fähigkeit auszeichnen, komplexe Probleme zu lösen, innovative Lösungen anzuwenden und Systeme und Datenbanken zu verwalten.

> *Sie werden ein vielseitiger Experte mit einer soliden technischen Ausbildung und strategischem Weitblick sein, der in der Lage ist, die Programmierung in technologische Lösungen zu integrieren, die den Bedürfnissen von Unternehmen und der Gesellschaft entsprechen.*

- **Kritisches Denken und Problemlösung:** Analyse komplexer Situationen und Erarbeitung innovativer und effizienter Lösungen durch Programmierung
- **Teamarbeit und multidisziplinäre Zusammenarbeit:** Arbeit in multidisziplinären Teams und effektive Zusammenarbeit mit Fachleuten aus verschiedenen Bereichen, wie Designern, Ingenieuren und Datenanalysten, um gemeinsame Ziele in Technologieprojekten zu erreichen
- **Zeitmanagement und selbstständiges Lernen:** Effektives Zeitmanagement, um knappe Fristen einzuhalten und Projekte selbstständig und verantwortungsbewusst durchzuführen
- **Anpassungsfähigkeit an neue Technologien:** Umgang mit neuen Tools und Programmiersprachen, wobei in einem sich rasant entwickelnden technologischen Umfeld eine ständige Aktualisierung gewährleistet wird

Nach Abschluss des Studiengangs werden Sie in der Lage sein, Ihre Kenntnisse und Fähigkeiten in den folgenden Positionen anzuwenden:

1. **Softwareentwickler:** Verantwortlich für die Gestaltung, Erstellung und Wartung von Anwendungen und Computerprogrammen entsprechend den Bedürfnissen des Kunden oder Unternehmens.
2. **Webentwickler:** Verantwortlich für die Erstellung, Gestaltung und Wartung von Websites, um deren Funktionalität und Optimierung auf verschiedenen Geräten sicherzustellen.
3. **Datenbankingenieur:** Verantwortlich für die Gestaltung, Implementierung und Verwaltung von Datenbanken, um deren Effizienz, Sicherheit und Verfügbarkeit zu gewährleisten.
4. **Systemanalytiker:** Entwickler bestehender IT-Systeme, der Verbesserungen zur Optimierung der Leistung und Effizienz der Prozesse vorschlägt
5. **Frontend-Programmierer:** Verantwortlich für die Entwicklung des visuellen Teils von Anwendungen und Websites, um die Benutzerfreundlichkeit zu verbessern
6. **Backend-Programmierer:** Verantwortlich für die Entwicklung der Serverlogik und der Datenbanken von Anwendungen und Websites, um deren ordnungsgemäßen Betrieb zu gewährleisten
7. **Netzwerkadministrator:** Verantwortlich für die Verwaltung und Wartung der Netzwerkinfrastruktur innerhalb eines Unternehmens, um deren Stabilität und Sicherheit zu gewährleisten.
8. **Technologieberater:** Berater und Ratgeber in Unternehmen, der Informationen über die Implementierung von technologischen Lösungen bereitstellt, die den Bedürfnissen des Unternehmens entsprechen und seine Leistung und Wettbewerbsfähigkeit verbessern.
9. **Entwickler für mobile Anwendungen:** Verantwortlich für das Design, die Entwicklung und Wartung von Anwendungen für mobile Geräte, um deren Benutzerfreundlichkeit und Leistung zu gewährleisten.
10. **Ingenieur für Automatisierungssoftware:** Verantwortlich für die Konzeption und Entwicklung automatisierter Lösungen zur Verbesserung der Effizienz von Geschäftsprozessen unter Verwendung von Programmierung und fortschrittlicher Technologie.

06

Studienmethodik

TECH ist die erste Universität der Welt, die die Methodik der ***case studies*** mit ***Relearning*** kombiniert, einem 100%igen Online-Lernsystem, das auf geführten Wiederholungen basiert.

Diese disruptive pädagogische Strategie wurde entwickelt, um Fachleuten die Möglichkeit zu bieten, ihr Wissen zu aktualisieren und ihre Fähigkeiten auf intensive und gründliche Weise zu entwickeln. Ein Lernmodell, das den Studenten in den Mittelpunkt des akademischen Prozesses stellt und ihm die Hauptrolle zuweist, indem es sich an seine Bedürfnisse anpasst und die herkömmlichen Methoden beiseite lässt.

TECH bereitet Sie darauf vor, sich neuen Herausforderungen in einem unsicheren Umfeld zu stellen und in Ihrer Karriere erfolgreich zu sein“

Der Student: die Priorität aller Programme von TECH

Bei der Studienmethodik von TECH steht der Student im Mittelpunkt.
Die pädagogischen Instrumente jedes Programms wurden unter Berücksichtigung der Anforderungen an Zeit, Verfügbarkeit und akademische Genauigkeit ausgewählt, die heutzutage nicht nur von den Studenten, sondern auch von den am stärksten umkämpften Stellen auf dem Markt verlangt werden.

Beim asynchronen Bildungsmodell von TECH entscheidet der Student selbst, wie viel Zeit er mit dem Lernen verbringt und wie er seinen Tagesablauf gestaltet, und das alles bequem von einem elektronischen Gerät seiner Wahl aus. Der Student muss nicht an Präsenzveranstaltungen teilnehmen, die er oft nicht wahrnehmen kann.
Die Lernaktivitäten werden nach eigenem Ermessen durchgeführt. Er kann jederzeit entscheiden, wann und von wo aus er lernen möchte.

> *Bei TECH gibt es KEINE Präsenzveranstaltungen (an denen man nie teilnehmen kann)“*

Die international umfassendsten Lehrpläne

TECH zeichnet sich dadurch aus, dass sie die umfassendsten Studiengänge im universitären Umfeld anbietet. Dieser Umfang wird durch die Erstellung von Lehrplänen erreicht, die nicht nur die wesentlichen Kenntnisse, sondern auch die neuesten Innovationen in jedem Bereich abdecken.

Durch ihre ständige Aktualisierung ermöglichen diese Programme den Studenten, mit den Veränderungen des Marktes Schritt zu halten und die von den Arbeitgebern am meisten geschätzten Fähigkeiten zu erwerben. Auf diese Weise erhalten die Studenten, die ihr Studium bei TECH absolvieren, eine umfassende Vorbereitung, die ihnen einen bedeutenden Wettbewerbsvorteil verschafft, um in ihrer beruflichen Laufbahn voranzukommen.

Und das von jedem Gerät aus, ob PC, Tablet oder Smartphone.

> *Das Modell der TECH ist asynchron, d. h. Sie können an Ihrem PC, Tablet oder Smartphone studieren, wo immer Sie wollen, wann immer Sie wollen und so lange Sie wollen“*

Case studies oder Fallmethode

Die Fallmethode ist das am weitesten verbreitete Lernsystem an den besten Wirtschaftshochschulen der Welt. Sie wurde 1912 entwickelt, damit Studenten der Rechtswissenschaften das Recht nicht nur auf der Grundlage theoretischer Inhalte erlernten, sondern auch mit realen komplexen Situationen konfrontiert wurden. Auf diese Weise konnten sie fundierte Entscheidungen treffen und Werturteile darüber fällen, wie diese zu lösen sind. Sie wurde 1924 als Standardlehrmethode in Harvard etabliert.

Bei diesem Lehrmodell ist es der Student selbst, der durch Strategien wie *Learning by doing* oder *Design Thinking*, die von anderen renommierten Einrichtungen wie Yale oder Stanford angewandt werden, seine berufliche Kompetenz aufbaut.

Diese handlungsorientierte Methode wird während des gesamten Studiengangs angewandt, den der Student bei TECH absolviert. Auf diese Weise wird er mit zahlreichen realen Situationen konfrontiert und muss Wissen integrieren, recherchieren, argumentieren und seine Ideen und Entscheidungen verteidigen. All dies unter der Prämisse, eine Antwort auf die Frage zu finden, wie er sich verhalten würde, wenn er in seiner täglichen Arbeit mit spezifischen, komplexen Ereignissen konfrontiert würde.

Relearning-Methode

Bei TECH werden die *case studies* mit der besten 100%igen Online-Lernmethode ergänzt: *Relearning*.

Diese Methode bricht mit traditionellen Lehrmethoden, um den Studenten in den Mittelpunkt zu stellen und ihm die besten Inhalte in verschiedenen Formaten zu vermitteln. Auf diese Weise kann er die wichtigsten Konzepte der einzelnen Fächer wiederholen und lernen, sie in einem realen Umfeld anzuwenden.

In diesem Sinne und gemäß zahlreicher wissenschaftlicher Untersuchungen ist die Wiederholung der beste Weg, um zu lernen. Aus diesem Grund bietet TECH zwischen 8 und 16 Wiederholungen jedes zentralen Konzepts innerhalb ein und derselben Lektion, die auf unterschiedliche Weise präsentiert werden, um sicherzustellen, dass das Wissen während des Lernprozesses vollständig gefestigt wird.

Das Relearning ermöglicht es Ihnen, mit weniger Aufwand und mehr Leistung zu lernen, sich mehr auf Ihre Spezialisierung einzulassen, einen kritischen Geist zu entwickeln, Argumente zu verteidigen und Meinungen zu kontrastieren: eine direkte Gleichung zum Erfolg.

Ein 100%iger virtueller Online-Campus mit den besten didaktischen Ressourcen

Um ihre Methodik wirksam anzuwenden, konzentriert sich TECH darauf, den Studenten Lehrmaterial in verschiedenen Formaten zur Verfügung zu stellen: Texte, interaktive Videos, Illustrationen und Wissenskarten, um nur einige zu nennen. Sie alle werden von qualifizierten Lehrkräften entwickelt, die ihre Arbeit darauf ausrichten, reale Fälle mit der Lösung komplexer Situationen durch Simulationen, dem Studium von Zusammenhängen, die für jede berufliche Laufbahn gelten, und dem Lernen durch Wiederholung mittels Audios, Präsentationen, Animationen, Bildern usw. zu verbinden.

Die neuesten wissenschaftlichen Erkenntnisse auf dem Gebiet der Neurowissenschaften weisen darauf hin, dass es wichtig ist, den Ort und den Kontext, in dem der Inhalt abgerufen wird, zu berücksichtigen, bevor ein neuer Lernprozess beginnt. Die Möglichkeit, diese Variablen individuell anzupassen, hilft den Menschen, sich zu erinnern und Wissen im Hippocampus zu speichern, um es langfristig zu behalten. Dies ist ein Modell, das als *Neurocognitive context-dependent e-learning* bezeichnet wird und in diesem Hochschulstudium bewusst angewendet wird.

Zum anderen, auch um den Kontakt zwischen Mentor und Student so weit wie möglich zu begünstigen, wird eine breite Palette von Kommunikationsmöglichkeiten angeboten, sowohl in Echtzeit als auch zeitversetzt (internes Messaging, Diskussionsforen, Telefondienst, E-Mail-Kontakt mit dem technischen Sekretariat, Chat und Videokonferenzen).

Darüber hinaus wird dieser sehr vollständige virtuelle Campus den Studenten der TECH die Möglichkeit geben, ihre Studienzeiten entsprechend ihrer persönlichen Verfügbarkeit oder ihren beruflichen Verpflichtungen zu organisieren. Auf diese Weise haben sie eine globale Kontrolle über die akademischen Inhalte und ihre didaktischen Hilfsmittel, in Übereinstimmung mit ihrer beschleunigten beruflichen Weiterbildung.

> *Der Online-Studienmodus dieses Programms wird es Ihnen ermöglichen, Ihre Zeit und Ihr Lerntempo zu organisieren und an Ihren Zeitplan anzupassen“*

Die Wirksamkeit der Methode wird durch vier Schlüsselergebnisse belegt:

1. Studenten, die diese Methode anwenden, nehmen nicht nur Konzepte auf, sondern entwickeln auch ihre geistigen Fähigkeiten durch Übungen zur Bewertung realer Situationen und zur Anwendung ihres Wissens.
2. Das Lernen basiert auf praktischen Fähigkeiten, die es den Studenten ermöglichen, sich besser in die reale Welt zu integrieren.
3. Eine einfachere und effizientere Aufnahme von Ideen und Konzepten wird durch die Verwendung von Situationen erreicht, die aus der Realität entstanden sind.
4. Das Gefühl der Effizienz der investierten Anstrengung wird zu einem sehr wichtigen Anreiz für die Studenten, was sich in einem größeren Interesse am Lernen und einer Steigerung der Zeit, die für die Arbeit am Kurs aufgewendet wird, niederschlägt.

Die von ihren Studenten am besten bewertete Hochschulmethodik

Die Ergebnisse dieses innovativen akademischen Modells lassen sich an der Gesamtzufriedenheit der Absolventen der TECH ablesen.

Die Studenten bewerten die pädagogische Qualität, die Qualität der Materialien, die Struktur und die Ziele der Kurse als ausgezeichnet. Es überrascht nicht, dass die Einrichtung im global score Index mit 4,9 von 5 Punkten die von ihren Studenten am besten bewertete Universität ist.

Sie können von jedem Gerät mit Internetanschluss (Computer, Tablet, Smartphone) auf die Studieninhalte zugreifen, da TECH in Sachen Technologie und Pädagogik führend ist.

Sie werden die Vorteile des Zugangs zu simulierten Lernumgebungen und des Lernens durch Beobachtung, d. h. Learning from an expert, nutzen können.

In diesem Programm stehen Ihnen die besten Lehrmaterialien zur Verfügung, die sorgfältig vorbereitet wurden:

Studienmaterial

Alle didaktischen Inhalte werden von den Fachkräfte, die den Kurs unterrichten werden, speziell für den Kurs erstellt, so dass die didaktische Entwicklung wirklich spezifisch und konkret ist.

Diese Inhalte werden dann auf ein audiovisuelles Format übertragen, das unsere Online-Arbeitsweise mit den neuesten Techniken ermöglicht, die es uns erlauben, Ihnen eine hohe Qualität in jedem der Stücke zu bieten, die wir Ihnen zur Verfügung stellen werden.

Übungen für Fertigkeiten und Kompetenzen

Sie werden Aktivitäten durchführen, um spezifische Kompetenzen und Fertigkeiten in jedem Fachbereich zu entwickeln. Übungen und Aktivitäten zum Erwerb und zur Entwicklung der Fähigkeiten und Fertigkeiten, die ein Spezialist im Rahmen der Globalisierung, in der wir leben, entwickeln muss.

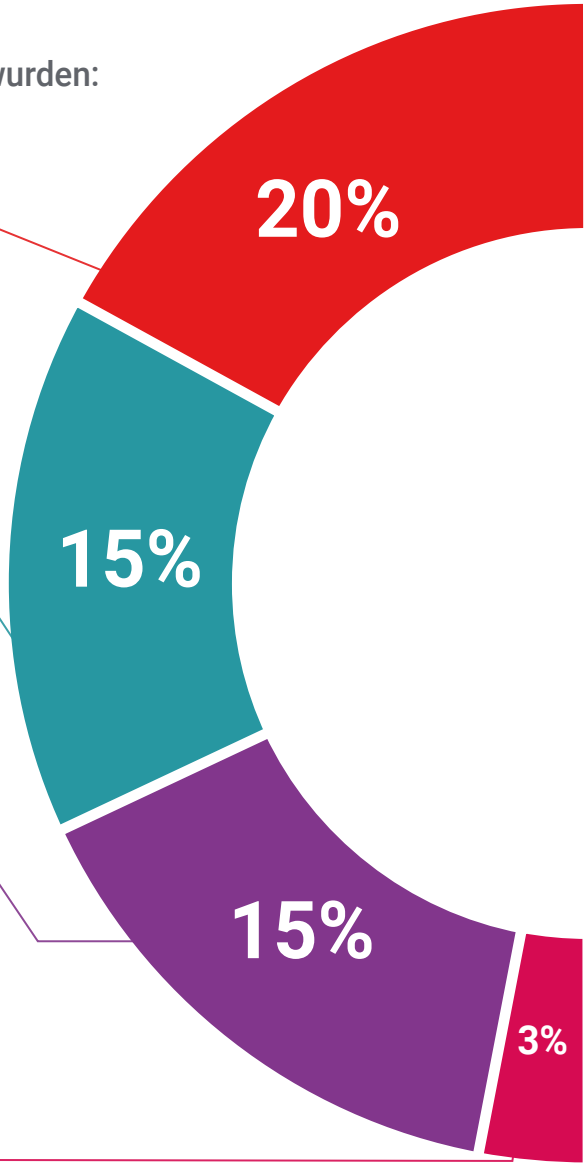
Interaktive Zusammenfassungen

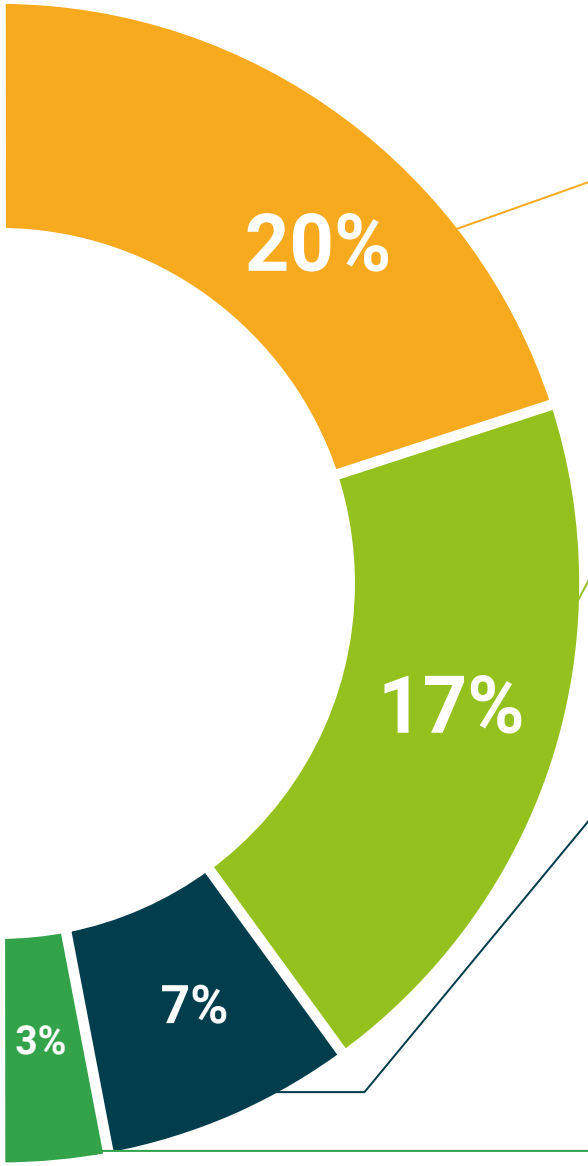
Wir präsentieren die Inhalte auf attraktive und dynamische Weise in multimedialen Pillen, Audios, Videos, Bilder, Diagramme und konzeptionelle Karten enthalten, um das Wissen zu festigen.

Dieses einzigartige System für die Präsentation multimedialer Inhalte wurde von Microsoft als „Europäische Erfolgsgeschichte" ausgezeichnet.

Weitere Lektüren

Aktuelle Artikel, Konsensdokumente, internationale Leitfäden... In unserer virtuellen Bibliothek haben Sie Zugang zu allem, was Sie für Ihre Ausbildung benötigen.





Case Studies

Sie werden eine Auswahl der besten *case studies* zu diesem Thema bearbeiten. Die Fälle werden von den besten Spezialisten der internationalen Szene präsentiert, analysiert und betreut.

Testing & Retesting

Während des gesamten Programms werden Ihre Kenntnisse in regelmäßigen Abständen getestet und wiederholt. Wir tun dies auf 3 der 4 Ebenen der Millerschen Pyramide.

Meisterklassen

Die Nützlichkeit der Expertenbeobachtung ist wissenschaftlich belegt.

Das sogenannte *Learning from an Expert* stärkt das Wissen und das Gedächtnis und schafft Vertrauen in unsere zukünftigen schwierigen Entscheidungen.

Kurzanleitungen zum Vorgehen

TECH bietet die wichtigsten Inhalte des Kurses in Form von Arbeitsblättern oder Kurzanleitungen an. Ein synthetischer, praktischer und effektiver Weg, um dem Studenten zu helfen, in seinem Lernen voranzukommen.

07

Lehrkörper

Der Lehrkörper besteht aus hochqualifizierten Fachleuten mit umfassender Erfahrung im Technologiebereich. Jeder einzelne wurde sorgfältig ausgewählt, um sicherzustellen, dass die Studenten eine qualitativ hochwertige Fortbildung erhalten, die auf aktuellem Wissen und der realen Praxis des Sektors basiert. In diesem Sinne ermöglichen ihre Berufserfahrung in renommierten Unternehmen und ihre akademische Ausbildung es ihnen, eine umfassende und angewandte Perspektive auf die Programmierung zu bieten. Kurz gesagt, die Mentoren sind Experten in verschiedenen Bereichen der Technologie, was eine bereichernde, multidisziplinäre Sichtweise ermöglicht.

Die Lehrkräfte sind eine wesentliche Säule für den Erfolg des Programms. Die Qualität und Erfahrung der Mentoren gewährleistet eine umfassende Fortbildung, die Sie darauf vorbereitet, dem Technologiesektor mit Selbstvertrauen und Kompetenz entgegenzutreten“

Leitung

Dr. Lucas Cuesta, Juan Manuel

- Senior Softwareingenieur und Analyst bei Indizen - Believe in Talent
- Senior Softwareingenieur und Analyst bei Krell Consulting und IMAGiNA Artificial Intelligence
- Softwareingenieur bei Intel Corporation
- Softwareingenieur bei Intelligent Dialogue Systems
- Promotion in elektronischer Systemtechnik für intelligente Umgebungen an der Polytechnischen Universität von Madrid
- Hochschulabschluss in Telekommunikationstechnik an der Polytechnischen Universität von Madrid
- Masterstudiengang in Elektroniksystemtechnik für intelligente Umgebungen an der Polytechnischen Universität von Madrid

Hr. Márquez Ruiz de Lacanal, Juan Antonio

- Softwareentwickler bei GTD Defence & Security Solutions
- Softwareentwickler bei Solera Inc.
- Entwicklungs- und Forschungsingenieur bei GRVC Sevilla
- Mitgründer von Unmute
- Mitgründer von VR Educa
- Akademischer Austausch in Ingenieurwesen und Unternehmertum an der University of California Berkeley
- Hochschulabschluss in Gesundheitstechnik an der Universität von Sevilla

Professoren

Dr. Luna Perejón, Francisco

- Spezialist für Computerarchitektur und -technologie
- Promotion in Computertechnik an der Universität von Sevilla
- Masterstudiengang in Computertechnik an der Universität von Sevilla
- Hochschulabschluss in Gesundheitstechnik an der Universität von Sevilla
- Hochschulabschluss in Computertechnik und Computertechnologien an der Universität Sevilla
- Mitglied von: Forschungsgruppe Robotik und Informationstechnologie (TEP108)

Hr. Péris Millán, Eduardo

- Direktor des technologischen Bereichs der Beratung
- Spezialist für Computertechnik
- Masterstudiengang Strategisches Management von Information und Wissen in Organisationen
- Masterstudiengang in Führung und öffentlicher Verwaltung
- Experte für Öffentliche Verwaltung
- Experte für Informatiksysteme für Smart CITIES

Hr. Pi Morell, Oriol

- Funktionsanalytiker bei Fihoca
- Product Owner für Hosting und E-Mail bei CDmon
- Funktionsanalytiker und Software Engineer bei Atmira und CapGemini
- Dozent bei CapGemini, CapGemini Forms und Atmira
- Hochschulabschluss in Technischem Ingenieurwesen in Computer Management an der Autonomen Universität von Barcelona
- Masterstudiengang in Künstliche Intelligenz an der Katholischen Universität von Avila
- Masterstudiengang MBA in Unternehmensführung und Verwaltung von IMF Smart Education
- Masterstudiengang in Management von Informationssystemen von IMF Smart Education
- Aufbaustudiengang in Design Patterns von der Offenen Universität von Katalonien (UOC)

Hr. Grillo Hernández, José Enrique

- Anwendungsentwickler und Technologieanalyst
- Senior Entwickler für mobile Anwendungen bei Globant
- Android-Entwickler bei Plexus Tech
- Senior Android-Entwickler bei RoadStr
- Leitender Entwickler für mobile Anwendungen bei Avantgarde IT-Information Technology Services
- Projektleiter bei Smartdess
- Entwickler bei Educatablet
- Technologie-Analyst bei Corporate Mobile Solutions
- Masterstudiengang in Systemtechnik an der Universität Simón Bolívar

Fr. Domínguez Valderrama, Desirée

- *Lead Product & Growth Strategist*
- Masterstudiengang in Grafikdesign und Kreativität an der Business School der Handelskammer von Sevilla
- Expertin für UX/UI-Designer von CoderHouse
- Expertin für Technologie und Unternehmertum

08

Qualifizierung

Der Privater Masterstudiengang in Programmierung von Grund auf garantiert neben der präzisesten und aktuellsten Fortbildung auch den Zugang zu einem von der TECH Global University ausgestellten Diplom.

Schließen Sie dieses Programm erfolgreich ab und erhalten Sie Ihren Universitätsabschluss ohne lästige Reisen oder Formalitäten"

Mit diesem Programm erwerben Sie den von **TECH Global University**, der größten digitalen Universität der Welt, bestätigten eigenen Titel **Privater Masterstudiengang in Programmierung von Grund auf**

TECH Global University ist eine offizielle europäische Universität, die von der Regierung von Andorra ***(Amtsblatt)*** öffentlich anerkannt ist. Andorra ist seit 2003 Teil des Europäischen Hochschulraums (EHR). Der EHR ist eine von der Europäischen Union geförderte Initiative, die darauf abzielt, den internationalen Ausbildungsrahmen zu organisieren und die Hochschulsysteme der Mitgliedsländer dieses Raums zu vereinheitlichen. Das Projekt fördert gemeinsame Werte, die Einführung gemeinsamer Instrumente und die Stärkung der Mechanismen zur Qualitätssicherung, um die Zusammenarbeit und Mobilität von Studenten, Forschern und Akademikern zu verbessern.

Dieser eigene Abschluss der **TECH Global University** ist ein europäisches Programm zur kontinuierlichen Weiterbildung und beruflichen Fortbildung, das den Erwerb von Kompetenzen in seinem Wissensgebiet garantiert und dem Lebenslauf des Studenten, der das Programm absolviert, einen hohen Mehrwert verleiht.

Titel: **Privater Masterstudiengang in Programmierung von Grund auf**

Modalität: **online**

Dauer: **12 Monate**

Akkreditierung: **60 ECTS**

tech global university

Hr./Fr. _____, mit der Ausweis-Nr. _____ hat erfolgreich bestanden und den folgenden Abschluss erworben:

Privater Masterstudiengang in Programmierung von Grund auf

Es handelt sich um einen eigenen Abschluss mit einer Dauer von 1.800 Stunden, was 60 ECTS entspricht, mit Anfangsdatum am dd/mm/aaaa und Enddatum am dd/mm/aaaa.

TECH Global University ist eine von der Regierung Andorras am 31. Januar 2024 offiziell anerkannte Universität, die dem Europäischen Hochschulraum (EHR) angehört.

Andorra la Vella, den 28. Februar 2024

Dr. Pedro Navarro Illana
Rektor

einzigartiger Code TECH: AFWOR23S techtitute.com/titel

Privater Masterstudiengang in Programmierung von Grund auf

Allgemeiner Aufbau des Lehrplans

Fachkategorie	Stunden
Obligatorisch (OB)	60
Wahlfach(OP)	0
Externes Praktikum (PR)	0
Masterarbeit (TFM)	0
	Summe 60

Allgemeiner Aufbau des Lehrplans

Kurs	Modul	ECTS	Kategorie
1º	Programmierung und Softwareentwicklung von Grund auf	6	OB
1º	Backend-Entwicklung I: Python von Grund auf	6	OB
1º	Backend-Entwicklung II – Algorithmen und Datenstrukturen mit Python von Grund auf	6	OB
1º	Backend-Entwicklung III – Flask, Erstellung von APIs und grundlegende Architektur von Grund auf	6	OB
1º	Objektorientierte Programmierung und Entwurfsmuster von Grund auf	6	OB
1º	Frontend I – HTML und CSS von Grund auf	6	OB
1º	Frontend II – JavaScript von Grund auf	6	OB
1º	Frontend III – React.js von Grund auf	6	OB
1º	Verwaltung und Optimierung von Datenbanken von Grund auf Tools für	6	OB
1º	die Entwicklung von Grund auf: Linux, Versionskontrolle, CI/CD, Docker und agile Methoden	6	OB

Dr. Pedro Navarro Illana
Rektor

tech global university

*Haager Apostille. Für den Fall, dass der Student die Haager Apostille für sein Papierdiplom beantragt, wird TECH Global University die notwendigen Vorkehrungen treffen, um diese gegen eine zusätzliche Gebühr zu beschaffen.

zukunft
gesundheit vertrauen menschen
erziehung information tutoren
garantie akkreditierung unterricht
institutionen technologie lernen
gemeinschaft verpflichtung
persönliche betreuung innovation
wissen gegenwart qualität
online-Ausbildung
entwicklung institutionen
virtuelles Klassenzimmer lernen

tech global university

Privater Masterstudiengang

Programmierung von Grund auf

- Modalität: **online**
- Dauer: **12 Monate**
- Qualifizierung: **TECH Global University**
- Akkreditierung: **60 ECTS**
- Zeitplan: **in Ihrem eigenen Tempo**
- Prüfungen: **online**

Privater Masterstudiengang
Programmierung von Grund auf

tech global university