

校级 硕士

从零开始编程



校级硕士 从零开始编程

- » 模式: 在线
- » 时长: 12个月
- » 学位: TECH Global University
- » 认证: 60 ECTS 学分
- » 课程表: 自由安排时间
- » 考试模式: 在线

网页链接: www.techtitute.com/cn/information-technology/master-degree/master-programming-scratch

目录

01

课程介绍

4

02

为什么在TECH学习?

8

03

教学大纲

12

04

教学目标

24

05

职业前景

30

06

学习方法

34

07

教学人员

44

08

学位

48

01 课程介绍

编程已成为 21 世纪的一项基本技能,在社会和经济的数字化转型中发挥着关键作用。根据OECD的一份报告,当今超过 60% 的工作需要高级数字技能,而编程是其中最紧缺的技能之一。考虑到编程是数字时代的一个重要领域,TECH 设计了这一校级硕士学位课程,教授该学科的基础知识及其在全球通用语言中的应用。通过 100% 在线学习方法,专家们将了解如何构建算法、管理数据和开发功能性项目,同时为在技术世界中继续前进打下坚实的基础。



“

了解如何编程从未如此容易。通过 TECH 自己的校级硕士学位,您将在任何地方掌握 Python 和 JavaScript 等语言,这要归功于 100%的在线方法。现在就报名吧!”

编程是现代社会正在经历的数字化转型的核心。从移动应用程序开发到业务流程自动化，了解如何编程是在不断变化的工作环境中取得成功的关键技能。从这个意义上说，掌握该领域的专业人士不仅会找到更好的成长机会，还会增强在技术不断发展的世界中适应和发展的能力。

意识到这一需求，TECH 设计了从零开始编程校级硕士，一个完整的课程，将提供最新的信息有关这一领域。该课程由行业专家设计，将涵盖从算法和数据结构的基本基础知识到 Python, JavaScript 和 HTML 等语言的的实际使用。还将包括网页设计、应用程序开发和数据库管理方面的专业模块。通过这种方式，专家将获得技术知识，并发展分析和创造性的技能，通过编程解决问题。所有这一切，与一个实际的方法，将保证实施所学到的概念。

通过专注于这一领域，毕业生不仅会在技术领域找到新的机会，而且会在金融、医疗保健或物流等行业脱颖而出，在这些行业中，编程发挥着关键作用。此外，他们将接受培训，领导开发项目，自动化流程和优化资源，提高他们的工作形象，并促进他们获得极具竞争力的出口。

最后，该课程的 100% 在线模式将提供必要的灵活性，将学习与其他个人或专业责任相结合。TECH 将为学生提供 24 小时开放的创新平台。此外，其 Relearning 方法将通过在不同环境中重申关键概念来优化培训，促进对内容的渐进和有效吸收。

这个**从零开始编程校级硕士学位**包含市场上最完整又最新的课程。主要特点是：

- 由编程专家介绍案例研究的发展情况
- 这个课程的内容图文并茂、示意性强、实用性强为那些视专业实践至关重要的学科提供了科学和实用的信息
- 可以进行自我评估过程的实践，以推进学习
- 特别强调创新的方法论
- 理论知识,专家预论,争议主题讨论论坛和个人反思工作
- 可以从任何有互联网连接的固定或便携式设备上获取内容

“

从基本算法到应用程序开发，这门课程将带你任何时间从零到专家。你将成为科技行业需要的专业人士!”

“

以无与伦比的议程进入编程的世界。通过市场上最全面的校级硕士学位课程,您将从零开始掌握 Python、JavaScript 和 Web 开发”

在技术专家教师的支持下,该学位将逐步指导您掌握市场上最受欢迎的语言。今天就开始,共创美好未来!

你会发现一个创新的方法,结合理论和实际项目在这个校级硕士学位。现在就报名吧,用代码将您的想法变为现实!

其教学人员包括来自程序领域的专业人士,他们将自己的工作经验带入这个课程,以及来自领先公司和著名大学的公认专家。

通过采用最新的教育技术制作的多媒体内容,专业人士将能够进行情境化学习,即通过模拟环境进行沉浸式培训,以应对真实情况。

这个课程的设计重点是基于问题的学习,通过这种方式,学生必须尝试解决整个学术课程中提出的不同专业实践情况。为此,职业人士将得到由著名专家开发的创新互动视频系统的协助。



02

为什么在TECH学习?

TECH 是世界上最大的数字大学。拥有超过14,000个大学课程的令人印象深刻的目录,涵盖11种语言,我们以就业率99%的领先地位跻身行业前列。此外,超过6,000名享有国际声誉的顶尖教授团队。



“

在世界上最大的数字大学学习并确保您的职业成功。未来始于TECH”

福布斯评选的全球最佳在线大学

著名的商业和金融杂志福布斯将泰晤士河科技大学评为《世界上最好的在线大学》。他们在数字版最近的一篇文章中提到了这一点，并在文中重复了这所学校的成功故事，«这要归功于它提供的学术课程，精选的师资队伍以及旨在培养未来专业人员的创新学习方法»

Forbes
世界上最好的
在线大学

更全面的
课程
计划

大学里最全面的学习计划

TECH 提供大学中最全面的课程，其主题涵盖基本概念以及特定科学领域的主要科学进步。这些课程也不断更新，以确保学生拥有最先进的学术技能和最需要的专业技能。通过这种方式，大学学位为毕业生在职业成功道路上提供了显著的优势。

最好的国际教学团队

TECH 的教学人员由 6,000 多名具有最高国际声望的教授组成。教授、研究人员和跨国公司高层管理人员，其中包括：Isaiah Covington，波士顿凯尔特人队的表现教练；Magda Romanska，哈佛MetaLAB的首席研究员；Ignacio Wistumba，MD安德森癌症中心转化分子病理学部门的主席；以及D.W Pine，TIME杂志的创意总监等。

国际
顶尖师资
团队

最有效的
教学方法

独特的学习方法

TECH 是第一所在所有学位中采用Relearning的大学。这是最好的在线学习方法，获得著名教育机构提供的国际教学质量认证。并且，这一颠覆性的学术模式与“案例教学法”相辅相成，构成了独特的在线教学策略。还提供创新的教材，包括详细的视频，信息图表和交互式摘要。

世界上最大的数字化大学

TECH 是世界上最大的数字大学。我们是最大的教育机构，拥有最好，最广泛的数字课程目录，100%在线且涵盖绝大多数知识领域。我们提供世界上最多的自主学位、官方研究生学位和本科学位。总共有超过 14,000 个大学学位，涵盖十种不同的语言，使我们成为世界上最大的教育机构。

世界排名第1
世界上第1大的
在线大学

NBA 官方在线大学

TECH是NBA的官方在线大学。由于与主要篮球联盟达成协议,该校为学生提供独家大学课程,以及专注于联盟业务和体育产业其他领域的各种教材。每个课程都有独特设计的课程设置,并邀请了杰出的演讲嘉宾:这些职业运动员具有卓越的运动经历,将分享他们在相关主题上的经验。

就业率领先者

TECH已成功成为就业能力领先的大学。99%的学生在完成大学课程后不到一年时间,就能在所学专业领域找到工作。同样多的人也成功地立即提升了自己的职业生涯。这一切都归功于一种学习方法,该方法的有效性基于掌握专业发展所必需的实践技能。



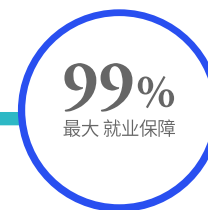
Google Partner Premier

这家北美科技巨头已向 TECH 授予Google Partner Premier 徽章。该奖项仅授予全球 3% 的公司,凸显了该大学为学生提供的有效,灵活和定制的体验。这一认可不仅认可了 TECH 数字基础设施的最高严谨性,性能和投资,而且还使该大学成为世界上最前沿的科技公司之一。



被学生评价为最佳大学

学生们将TECH定位为世界上在主要评价平台上评价最高的大学,突出其获得的4.9分(满分5分)的高评价,这一评分来自于超过1,000条评论。这些成绩巩固了TECH作为国际标杆大学机构的地位,反映了其教育模式的卓越性和积极影响。



03 教学大纲

在整个课程中, 专业人员将学习算法、数据结构和编程逻辑的基础知识, 然后学习Python, JavaScript 和 HTML等语言。此外, 他们还将深入研究网页设计、应用程序开发和数据库管理等关键领域, 提供编程世界的全面概述。由于采用了渐进式和结构化的方法, 该课程不仅将确保获得技术知识, 而且还将培养在竞争激烈的工作环境中不可或缺的分析 and 创造能力。



“

这个研究生学位是理想的选择，
开始你的职业生涯在技术部门，
并获得必要的工具，以脱颖而出，
在任何专业领域相关的编程”

模块1.从零开始的软件编程与开发

- 1.1. 软硬件。关系与差异
 - 1.1.1. 软件
 - 1.1.2. 软件和硬件之间的差异
 - 1.1.3. 软件、硬件和编程之间的关系
- 1.2. 编程。关键问题
 - 1.2.1. 编程
 - 1.2.2. 目标和应用
 - 1.2.3. 程序、源代码、编译和执行
 - 1.2.4. 误差: 语法、执行和逻辑错误
- 1.3. 从零开始编程
 - 1.3.1. 程序的结构
 - 1.3.2. 算法: 结构和例子
 - 1.3.3. 算法与程序的关系
 - 1.3.4. 使用算法解决问题
- 1.4. 编程语言的范式和类型
 - 1.4.1. 编程范式
 - 1.4.1.1. 命令式范式
 - 1.4.1.2. 面向对象范式
 - 1.4.1.3. 函数式范式
 - 1.4.1.4. 声明式范式
 - 1.4.2. 低级语言与高级语言
 - 1.4.3. 编译语言与解释语言的区别
- 1.5. 编程语言的翻译方式
 - 1.5.1. 编译器。编译过程
 - 1.5.2. 解释器: 解释过程
 - 1.5.3. 编译与解释的区别
- 1.6. 比特、二进制运算与逻辑门
 - 1.6.1. 比特二进制表示法
 - 1.6.2. 位运算基础: AND, OR, XOR, NOT
 - 1.6.3. 二进制与十进制之间的转换
 - 1.6.4. 逻辑门: OR, AND, XOR, NOT, NOR 和 NAND





- 1.7. 从零开始设计算法
 - 1.7.1. 从零开始设计算法
 - 1.7.2. 顺序、条件与循环算法
 - 1.7.3. 递归算法及其与迭代算法的比较
- 1.8. 程序的组成元素:组件与结构
 - 1.8.1. 输入和输出数据
 - 1.8.2. 变量和常量:用途
 - 1.8.3. 数据处理与操作
 - 1.8.4. 基本函数与过程,配合流程图表示
- 1.9. 控制结构与流程图
 - 1.9.1. 控制结构。在编程中的作用
 - 1.9.2. 条件结构示例含流程图
 - 1.9.3. 循环结构示例示例含流程图
- 1.10. 软件生命周期与开发模型
 - 1.10.1. 软件的生命周期。阶段
 - 1.10.2. 开发模型:瀑布模型、迭代模型与敏捷模型
 - 1.10.3. 软件开发中的测试与维护。

模块2.Backend 开发 I: Python 从头开始

- 2.1. Python 从头开始安装
 - 2.1.1. Python语言。特点
 - 2.1.2. 在 Windows、macOS 和 Linux 上安装 Python
 - 2.1.3. 开发环境配置:集成开发环境和代码编辑器
 - 2.1.4. Python 中的第一个程序:“哈喽世界”
- 2.2. Python 语法和变量
 - 2.2.1. Python 代码结构:标识
 - 2.2.2. Python 中的注释
 - 2.2.3. Python 中的变量和数据类型
 - 2.2.4. Python 中的算术和逻辑运算
- 2.3. 流量控制:条件
 - 2.3.1. 控制结构
 - 2.3.2. 条件语句:if, elif, else
 - 2.3.3. 三元条件表达式

- 2.4. Python中的循环
 - 2.4.1. 编程中循环的使用
 - 2.4.2. “for”和“while”循环
 - 2.4.3. 循环中的流程控制:break和continue
 - 2.4.4. 嵌套循环
- 2.5. Python中的函数
 - 2.5.1. Python中的函数。用途
 - 2.5.2. 函数的参数和实参
 - 2.5.3. 返回值
 - 2.5.4. 预定义函数与用户自定义函数
- 2.6. Python中的列表和元组
 - 2.6.1. 在Python中创建和使用列表
 - 2.6.2. 列表的常见操作:添加、删除、修改
 - 2.6.3. 元组:与列表的区别
 - 2.6.4. 对列表和元组的迭代
- 2.7. Python中的字典和集合
 - 2.7.1. 字典:键-值对
 - 2.7.2. 操作字典的方法
 - 2.7.3. 集合:用途
 - 2.7.4. 字典和集合的比较。
- 2.8. 从零开始的Python文件操作
 - 2.8.1. 文件的打开和关闭
 - 2.8.2. 打开模式:读取、写入和追加
 - 2.8.3. 文本文件的读写
- 2.9. 错误和异常处理
 - 2.9.1. 异常类型
 - 2.9.2. 使用Try、Except来管理错误
 - 2.9.3. 创建自定义异常
- 2.10. Python中的最佳实践和调试
 - 2.10.1. Debugging: 目的
 - 2.10.2. 调试技术:使用print和断点。
 - 2.10.3. 编写代码的最佳实践

模块3.Backend开发 II -用 Python 从零开始学习算法和数据结构

- 3.1. 数据结构中的搜索算法
 - 3.1.1. 数据搜索算法的目的
 - 3.1.2. 线性搜索:实现和使用场景
 - 3.1.3. 二分搜索:实例
 - 3.1.4. 效率比较:线性搜索与二分搜索
- 3.2. 数据结构中的排序算法 (I)。基础技术,如冒泡排序和插入排序
 - 3.2.1. 冒泡排序(Bubble Sort):实现和分析
 - 3.2.2. 插入排序(Insertion Sort):实现和使用场景
 - 3.2.3. 冒泡排序与插入排序的比较
- 3.3. 数据结构中的排序算法 (II)。高级技术,如选择排序、归并排序和快速排序
 - 3.3.1. Selection Sort.实现和分析
 - 3.3.2. Merge Sort: 实施
 - 3.3.3. Quick Sort: 实施
 - 3.3.4. 排序算法的效率比较
- 3.4. 数据结构中的递归搜索算法
 - 3.4.1. 递归用途
 - 3.4.2. 直接递归与间接递归
 - 3.4.3. 递归算法示例:阶乘和斐波那契数列实例
- 3.5. 数据搜索算法的复杂度分析
 - 3.5.1. 算法复杂度:效率度量
 - 3.5.2. 大O符号表示法
 - 3.5.3. 搜索和排序算法的复杂度分析
- 3.6. 高级数据结构
 - 3.6.1. 树:术语
 - 3.6.2. 二叉树:操作
 - 3.6.3. 二叉搜索树(BST):搜索、插入和删除操作
- 3.7. 图算法
 - 3.7.1. 图表代表
 - 3.7.2. 图的算法:DFS和BFS
 - 3.7.3. DFS和BFS之间比较

- 3.8. 动态编程
 - 3.8.1. 动态编程用处
 - 3.8.2. 动态编程与递归编程的区别
 - 3.8.3. 通过动态编程进行优化
- 3.9. 数据搜索算法的优化技术
 - 3.9.1. 优化数据搜索算法的重要性
 - 3.9.2. 优化技术: 记忆化技术 (Memoization)
 - 3.9.3. Divide and Conquer: 区分与你将克服
- 3.10. Python中的其他算法
 - 3.10.1. 排列与组合算法
 - 3.10.2. 基本哈希算法
 - 3.10.3. 子集计数与生成算法
- 4.4. APIs RESTful
 - 4.4.1. API RESTful.原则
 - 4.4.2. RESTful API 中的 HTTP 方法和约定
 - 4.4.3. 在 Flask 中创建 RESTful API
 - 4.4.4. 设计具有 CRUD 操作的任务 API
- 4.5. 数据库和 Flask 与 SQLite
 - 4.5.1. Web 应用程序中的数据库
 - 4.5.2. 在 Flask 项目中连接到 SQLite
 - 4.5.3. 使用 SQLAlchemy 创建表和模型
 - 4.5.4. SQLite 中的 CRUD 查询用于数据管理

模块4.Backend开发 III - Flask, 从零开始创建 API 和基本架构

- 4.1. Flask 作为 Framework Backend
 - 4.1.1. Framework Backend.目的
 - 4.1.2. Flask.特点
 - 4.1.3. 准备开发环境并安装Flask
 - 4.1.4. 使用 Flask 的第一个项目:“哈喽世界”
- 4.2. Flask 中的 HTTP 路由和 HTTP 请求
 - 4.2.1. 路由: 在 Web 应用程序中操作
 - 4.2.2. Flask 中的 HTTP 方法:GET, POST, PUT 和 DELETE
 - 4.2.3. Flask 中带有参数和数据的路由
 - 4.2.4. 组织项目中的路由
- 4.3. Flask 中的控制器和响应处理
 - 4.3.1. 控制器: 作用和责任
 - 4.3.2. Flask 中的响应类型文本、JSON 和 HTML
 - 4.3.3. 在 Flask 中为 API 创建控制器
 - 4.3.4. 控制器中的 CRUD 操作
- 4.6. API 中的身份验证和基本安全
 - 4.6.1. API 中的身份验证和授权
 - 4.6.2. 创建用户身份验证系统
 - 4.6.3. 在 Flask 中使用令牌进行身份验证
 - 4.6.4. 保护 API 中的路由和用户数据
- 4.7. 数据验证和错误处理
 - 4.7.1. Flask 中的错误和异常处理
 - 4.7.2. API 请求中的数据验证
 - 4.7.3. 创建自定义错误消息
 - 4.7.4. CRUD 中的验证和错误处理策略
- 4.8. 构建可扩展的 API
 - 4.8.1. 可扩展 Flask 项目的组织和结构
 - 4.8.2. API 的模块化和职责分离
 - 4.8.3. 针对性能和可扩展性的基本 API 优化
 - 4.8.4. 大型项目的组织策略
- 4.9. 使用 WebSocket 进行实时通信
 - 4.9.1. WebSockets.应用
 - 4.9.2. 使用 Flask-SocketIO 在 Flask 中实现 WebSocket
 - 4.9.3. Flask 应用程序中的实时通信

- 4.10. 应用程序部署和维护
 - 4.10.1. 准备 Flask 应用以供生产
 - 4.10.2. 部署到 Heroku 和 Render 等流行平台
 - 4.10.3. 使用 Docker 进行容器部署
 - 4.10.4. backend应用程序的监控和维护

模块5.从零开始的面向对象编程和设计模式

- 5.1. 面向对象编程 (OOP) 从零开始
 - 5.1.1. 面向对象程序设计
 - 5.1.2. 面向对象编程与结构化编程的区别
 - 5.1.3. 面向对象的基本元素:类、对象、方法与属性
- 5.2. Python中的类与对象
 - 5.2.1. 在Python中创建类和对象
 - 5.2.2. 实例属性与类属性
 - 5.2.3. 特殊方法 (init, str, repr等)
 - 5.2.4. 静态方法与类方法的使用用途
- 5.3. 类中的封装与抽象
 - 5.3.1. 封装:用途
 - 5.3.2. Python中的访问修饰符
 - 5.3.2.1. (公共、受保护、私有)
 - 5.3.3. 抽象:隐藏实现细节,简化接口
 - 5.3.4. 使用 (@property) 属性控制访问权限
- 5.4. Python中的继承及其在OOP中的作用
 - 5.4.1. 继承:在OOP中的作用
 - 5.4.2. 创建派生类与多重继承
 - 5.4.3. 方法和属性的继承与重载
 - 5.4.4. 类的层次结构与基类管理
- 5.5. Python中的多态与重载
 - 5.5.1. 多态性: Duck Typing
 - 5.5.2. 基于类和方法的多态实现
 - 5.5.3. 方法的重载与重写
 - 5.5.4. 软件设计中的多态:应用与优势
- 5.6. 类之间的关系与复杂结构设计
 - 5.6.1. 类关系类型:关联、聚合与组合
 - 5.6.2. 聚合与组合的区别及示例实例
 - 5.6.3. 基于类关系设计复杂系统结构
- 5.7. 设计模式与SOLID原则
 - 5.7.1. 设计模式的重要性
 - 5.7.2. 在OOP项目中的设计模式应用与优势优势
 - 5.7.3. 设计模式分类。
 - 5.7.4. SOLID设计原则及其在OOP中的作用
- 5.8. 创建型设计模式
 - 5.8.1. 创建型设计模式的目的
 - 5.8.2. Singleton模式
 - 5.8.3. Factory模式和 Factory Method
 - 5.8.4. 构建器模式
- 5.9. 结构设计模式
 - 5.9.1. 结构型设计模式的目的
 - 5.9.2. 适配器模式
 - 5.9.3. 装饰器模式
 - 5.9.4. 外观模式
- 5.10. 行为型设计模式
 - 5.10.1. 行为型设计模式及其应用
 - 5.10.2. Observer模式
 - 5.10.3. Strategy模式

模块6.Frontend I - 从头开始学习 HTML 和 CSS

- 6.1. 从头开始学习 HTML
 - 6.1.1. HTMLWeb开发的目的
 - 6.1.2. HTML 文档的结构:DOCTYPE, <html>, <head>, <body>
 - 6.1.3. 语义和内容标签:<header>, <nav>, <section>, <footer>
 - 6.1.4. 基本要素:段落 (<p>)、列表 (、)、链接 (<a>)、
图片 ()
 - 6.1.5. HTML 中的良好做法
- 6.2. HTML 中的文本和多媒体元素
 - 6.2.1. 基本文本标签:<p>、标题、列表、粗体和斜体
 - 6.2.2. 插入多媒体:、<audio>、<video> 的属性。
 - 6.2.3. 可访问性的基本属性 (alt、aria-label)
- 6.3. HTML表单
 - 6.3.1. 表单结构和组件:<form>, <input>, <label>, <button>
 - 6.3.2. 输入类型:文本、电子邮件、密码、提交按钮
 - 6.3.3. HTML5验证:客户端字段验证
 - 6.3.4. 具有基本验证功能的表单。实例
- 6.4. 从头开始学习 CSS
 - 6.4.1. 从头开始学习 CSS 语言:与 HTML 的使用和关系
 - 6.4.2. CSS语法:选择器、属性和值
 - 6.4.3. 应用内联、内部和外部样式
 - 6.4.4. 高阶选择器:类型、类、ID、伪类 (:hover、:focus)
- 6.5. CSS中的盒子模型(Box Model)
 - 6.5.1. 盒子模型: CSS的重要性
 - 6.5.2. 关键属性:边距、填充、边框、宽度、高度
 - 6.5.3. 使用 box-sizing 来精确控制盒子模型
 - 6.5.4. 该设计应用了盒子模型。实例
- 6.6. CSS 中的排版和文本样式
 - 6.6.1. 颜色和字体属性:颜色、字体系列、字体大小
 - 6.6.2. 高级文本样式:粗体、斜体、对齐(文本对齐)
 - 6.6.3. 文本间距和分隔:行高、字母间距
 - 6.6.4. CSS 中的测量单位(px、em、rem) 及其在排版中的使用

- 6.7. 使用 CSS 设计布局 - Flexbox
 - 6.7.1. Flexbox: 目的
 - 6.7.2. Flexbox框属性:justify-content、align-items、flex-direction
 - 6.7.3. Flexbox 中的元素布局和对齐
 - 6.7.4. Flexbox 布局示例
- 6.8. CSS 网格和 CSS 响应式设计
 - 6.8.1. CSS Grid: 行、列和区域
 - 6.8.2. Media Queries: 不同设备上的结构和应用
 - 6.8.3. 针对移动设备、平板电脑和台式电脑的响应式设计
 - 6.8.4. 响应式设计中的排版和流体单位调整
- 6.9. CSS 中的动画和过渡
 - 6.9.1. 过渡: 过渡属性, 对 :hover 的影响
 - 6.9.2. 使用 CSS 的动画:使用@keyframes, 基本动画
 - 6.9.3. 网络上平滑过渡和动画的技术
- 6.10. 设计中的网页可访问性
 - 6.10.1. 网络可访问性:重要性
 - 6.10.2. 可访问的网站设计。最佳实践
 - 6.10.3. Aria 标签和可访问性验证工具

模块7.Frontend II - 从零开始学习 JavaScript

- 7.1. 从头开始学习 JavaScript
 - 7.1.1. JavaScript 语言
 - 7.1.2. 将 JavaScript 集成到 HTML
 - 7.1.3. JavaScript 中的第一个程序:“哈喽世界”
- 7.2. JavaScript 中的变量和数据类型
 - 7.2.1. 使用 var、let 和 const 声明变量
 - 7.2.2. 数据类型:数字、字符串、布尔值
 - 7.2.3. 数据类型之间的转换
- 7.3. JavaScript 中的控制结构
 - 7.3.1. 条件:if、else if、else
 - 7.3.2. 循环:for、while、do...while
 - 7.3.3. Switch-case: 多个条件的替代方案
 - 7.3.4. 循环中的 break 和 continue

- 7.4. JavaScript 中的函数
 - 7.4.1. 函数声明
 - 7.4.2. 参数、返回值和范围
 - 7.4.3. 箭头函数 (=>) 和匿名函数
 - 7.4.4. 函数中的回调和递归
- 7.5. 使用 JavaScript 操作 DOM(Document Object Model)
 - 7.5.1. DOM: HTML 文档结构
 - 7.5.2. 选择 DOM 元素 (getElementById、querySelector)
 - 7.5.3. 元素操作:更改文本、样式和属性
 - 7.5.4. 事件:点击、输入、提交等
- 7.6. JavaScript 中的数组和对象
 - 7.6.1. 声明和使用数组
 - 7.6.2. 常用数组方法:push、pop、map、filter
 - 7.6.3. 创建和使用对象
 - 7.6.4. 迭代数组和对象
- 7.7. JavaScript 中的 Promise 和异步
 - 7.7.1. JavaScript 中的异步和回调的使用
 - 7.7.2. JavaScript 中的 Promise:创建和管理
 - 7.7.3. 在 JavaScript 中使用 async 和 await
- 7.8. JavaScript 中的 API 和 Fetch
 - 7.8.1. JavaScript 中的 API:目的
 - 7.8.2. 使用 fetch 来使用 REST API
 - 7.8.3. 处理错误和请求状态
- 7.9. Web 浏览器中的本地存储
 - 7.9.1. Web 服务和应用程序中的本地存储和会话存储
 - 7.9.2. 在本地存储中存储和检索数据
 - 7.9.2. IndexedDB 作为浏览器的数据库
 - 7.9.3. 在 JavaScript 中处理 Cookie
 - 7.9.4. 浏览器存储:实例
- 7.10. JavaScript 最佳实践和开发者工具
 - 7.10.1. JavaScript代码:最佳实践
 - 7.10.2. 使用 JavaScript 中的浏览器开发工具
 - 7.10.3. JavaScript 中的调试和错误处理

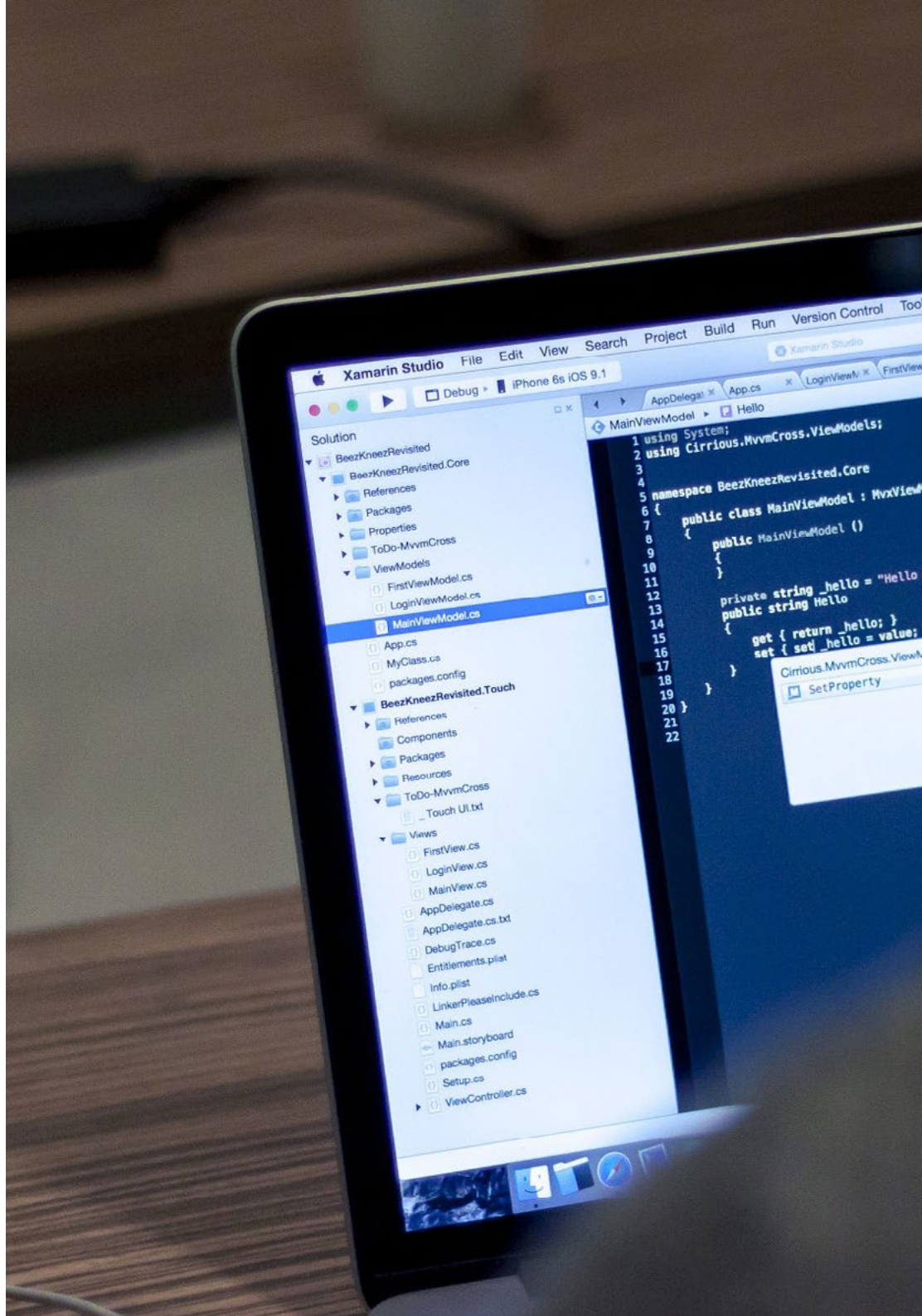
模块8.Frontend III - 从头开始使用 React.js

- 8.1. 从头开始使用 React.js
 - 8.1.1. React JS 作为开发 Web 应用程序的库
 - 8.1.2. React JS 中的组件和虚拟 DOM:架构与操作
 - 8.1.3. 使用 NextJS 框架进行安装和配置
 - 8.1.4. React 中的第一个组件:“哈喽世界”
- 8.2. JavaScript XML 或 JSX, 以及 React 中的组件
 - 8.2.1. JSX: 语法和特征
 - 8.2.2. 在 React.js 中创建函数组件
 - 8.2.3. 使用 props 在组件之间传递数据
 - 8.2.4. React.js 开发中的函数组件与类组件
- 8.3. React.js 中的状态和事件
 - 8.3.1. React 中的组件状态
 - 8.3.2. 使用 useState 进行状态管理
 - 8.3.3. React.js 中的事件处理:onClick、onChange 等
 - 8.3.4. React.js 中的状态和事件管理示例
- 8.4. React 中的组件和效果生命周期
 - 8.4.1. React 组件生命周期
 - 8.4.2. 使用 useEffect 处理 React 中的效果
 - 8.4.3. React 中具有挂载、更新和卸载功能的组件
- 8.5. 使用 React Router 进行路由
 - 8.5.1. SPA(Single Page Applications)和 Web 应用程序中的路由
 - 8.5.2. 安装和配置 React Router
 - 8.5.3. 使用 React Router 创建路由并在页面之间导航
- 8.6. React 中的表单和验证
 - 8.6.1. 在 React 中创建交互式表单
 - 8.6.2. 在 React 中处理用户输入并提交数据
 - 8.6.3. React 中的实时表单验证
- 8.7. 在 React 中使用 API
 - 8.7.1. 在 React 中使用 fetch 和 axios 调用 API
 - 8.7.2. 在 React 中处理加载、成功和错误状态
 - 8.7.3. 在 React 中根据 API 数据更新组件

- 8.8. React 中的可重用组件和外部库
 - 8.8.1. React 中的可重用组件
 - 8.8.2. 在 React 中创建可重用组件
 - 8.8.3. 在 React 中使用外部库例如 Material UI 和 Bootstrap
 - 8.9. React 中的全局状态管理
 - 8.9.1. 具有本机选项的全局状态管理: Context API 和自定义 Hooks
 - 8.9.2. 用于数据管理的外部库
 - 8.9.3. 全局状态管理方法的比较。实例
 - 8.10. 部署和优化 React 应用程序
 - 8.10.1. 准备用于生产的 React 应用程序
 - 8.10.2. 在 Netlify 和 Vercel 等平台上部署
 - 8.10.3. 性能优化: 延迟加载、记忆、服务器组件和代码拆分。
 - 8.10.4. 监控和维护生产中的 React 应用程序。工具和性能分析
- ## 模块9.从零开始的数据库管理与优化
- 9.1. 从零开始的数据库
 - 9.1.1. 数据库: 类型
 - 9.1.2. 关系数据库与非关系数据库
 - 9.1.3. SQL 和 NoSQL 编程语言
 - 9.2. 关系数据建模
 - 9.2.1. 关系数据库模型
 - 9.2.2. 关系数据库中的表、行和列
 - 9.2.3. 主键和外键: 表之间的关系
 - 9.2.4. 规范化: 1NF, 2NF, 3NF
 - 9.3. SQL 语言: DML 和 DDL
 - 9.3.1. SQL: 结构化查询语言
 - 9.3.2. 创建和删除查询: CREATE, DROP
 - 9.3.3. SELECT, INSERT, UPDATE 和 DELETE 查询
 - 9.3.4. 使用 SQL 进行数据筛选与排序
 - 9.4. SQL 高级查询
 - 9.4.1. Joins: INNER JOIN y OUTER JOIN
 - 9.4.2. 子查询与嵌套查询
 - 9.4.3. SQL 中的聚合函数: SUM, AVG, COUNT
 - 9.5. NoSQL 数据库和 MongoDB
 - 9.5.1. NoSQL 数据库
 - 9.5.2. SQL 与 NoSQL 的比较
 - 9.5.3. MongoDB: 文档数据库
 - 9.5.4. NoSQL 中的灵活模式
 - 9.6. 数据库优化
 - 9.6.1. 查询优化的重要性
 - 9.6.2. 在关系数据库中使用索引
 - 9.6.3. NoSQL 数据库优化
 - 9.7. 数据库安全
 - 9.7.1. 数据库安全
 - 9.7.2. 敏感数据加密
 - 9.7.3. 管理数据库中的用户和权限
 - 9.7.4. 保护数据库免受攻击的策略
 - 9.8. 数据库可扩展性
 - 9.8.1. 数据库可扩展性
 - 9.8.2. 水平和垂直分区
 - 9.8.3. 数据库中的复制和集群
 - 9.9. 数据备份与恢复
 - 9.9.1. 数据库备份的重要性
 - 9.9.2. 自动和手动备份技术
 - 9.9.3. 关系型数据库与 NoSQL 数据库的数据恢复
 - 9.10. 项目中的数据库实现
 - 9.10.1. 实际项目中的数据库设计
 - 9.10.2. 数据库与后端应用的集成

模块10.从头开始的开发工具:Linux、版本控制、CI/CD、Docker 和敏捷方法

- 10.1. 从零开始学习 Linux
 - 10.1.1. Linux
 - 10.1.2. Linux 与其他操作系统的区别
 - 10.1.3. 开发者常用的流行 Linux 发行版
 - 10.1.4. 开发环境的配置与个性化
 - 10.1.5. Linux 中的文本编辑器
- 10.2. 从零开始使用 Linux 终端
 - 10.2.1. 终端的用途与功能
 - 10.2.2. 文件导航与管理命令
 - 10.2.3. Linux 中的文件和目录权限
 - 10.2.4. 命令重定向与管道(pipes)的优化使用
- 10.3. 从零开始使用 Git 进行版本控制
 - 10.3.1. Git: 云服务提供商
 - 10.3.2. 仓库的创建与管理
 - 10.3.3. 工作流程: git init, git add, git commit 和 git status
 - 10.3.4. 分支管理:创建、合并与冲突解决
- 10.4. 从零开始的 GitHub 团队协作
 - 10.4.1. GitHub: 远程存储库
 - 10.4.2. 本地仓库与 GitHub 的连接: git remote.配置
 - 10.4.3. 与远程仓库的同步
 - 10.4.4. Pull Request 与协作式代码审查
- 10.5. CI/CD (I) – 从零开始使用 GitHub Actions 进行持续集成 (CI)
 - 10.5.1. 持续集成 (CI) 简介
 - 10.5.2. GitHub Actions 的工作流配置
 - 10.5.3. 测试与部署自动化
- 10.6. 从零开始使用 Docker
 - 10.6.1. Docker 与容器技术
 - 10.6.2. Docker 的安装与配置
 - 10.6.3. Docker 容器的创建与管理
 - 10.6.4. Dockerfile:自定义镜像的创建





- 10.7. CI/CD (II) – 从零开始使用 Docker 和 GitHub Actions 实现持续交付 (CD)
 - 10.7.1. 持续交付 (CD) 简介
 - 10.7.2. 使用 Docker 和 GitHub Actions 配置 CD 流水线
 - 10.7.3. 使用 Docker Compose 实现自动化部署
- 10.8. 从零开始的敏捷方法论 (I)。原则与价值
 - 10.8.1. 敏捷方法: 原则
 - 10.8.2. 敏捷宣言: 核心价值与原则
 - 10.8.3. 与传统方法对比: 瀑布式与敏捷式
- 10.9. 敏捷方法 (II): 从零开始学习 Scrum
 - 10.9.1. Scrum 及其适用性
 - 10.9.2. Scrum 中的关键角色: Product Owner, Scrum Master 和 Development Team
 - 10.9.3. Scrum 工件: Product Backlog, Sprint Backlog 和增量积压
 - 10.9.4. Scrum 活动: Sprint Planning, Daily Scrum, Sprint Review 和 Retrospective
- 10.10. 敏捷方法 (III): 从零开始学习 Kanban 与敏捷指标
 - 10.10.1. Kanban 及其可视化方法
 - 10.10.2. Kanban 的关键元素: 列、卡片与 WIP 限制
 - 10.10.3. 敏捷指标: Burnup, burndown charts, velocity 和 lead tim

“

您将在舒适的家中按照自己的节奏进行培训, 并获得所需的创新在线方法和灵活性。不要再等待了, 在技术领域提升您的职业生涯! 今天就报名吧”

04

教学目标

这个校级硕士学位本身的主要目标是为专业人士提供必要的技能,以坚实和有效的方式开始他们在编程世界的职业生涯。在整个课程中,他们将深入了解编程的基本概念,如算法、数据结构和编程逻辑,这是他们建立更高级技能的基础。此外,他们还将掌握行业中广泛使用的编程语言(Python, JavaScript和HTML),使他们能够自信地承担各种技术项目。

“

您将获得领导开发项目、管理数据库和优化流程所需的技能,从而提高您在技术领域的专业竞争力”



总体目标

- ◆ 通过识别程序的基本元素、控制结构以及软件和硬件之间的关系,掌握编程和软件开发的基础知识
- ◆ 了解 Python 的基础知识及其基本语法,开发适合backend开发的环境
- ◆ 深入了解基本算法和数据结构
- ◆ 掌握backend开发的基础知识及其在软件体系结构中的作用
- ◆ 管理面向对象编程的原则,并将其应用于使用 Python 作为编程语言构建灵活、可重用和可维护的软件
- ◆ 开发 HTML 和 CSS 的基本概念,以构建和设置网页的样式
- ◆ 分析 JavaScript 语言,从基础知识到高级技术
- ◆ 介绍 React.js 构建交互式 Web 应用程序的基本概念
- ◆ 在关系和 NoSQL 环境中管理数据库类型及其查询语言
- ◆ 详细说明测试自动化、集成和代码部署





具体目标

模块1.从零开始的软件编程与开发

- ◆ 定义并区分软件和硬件
- ◆ 奠定基本编程概念的基础
- ◆ 了解程序的基本结构
- ◆ 探索和分析不同的编程范式

模块2.Backend 开发 I: Python 从头开始

- ◆ 掌握 Python 特性
- ◆ 了解Python的基本结构和语法
- ◆ 培养使用条件控制流程的技能
- ◆ 在 Python 中应用循环来创建重复循环

模块3.Backend开发 II -用 Python 从零开始学习算法和数据结构

- ◆ 实现并比较数据结构中的搜索算法类型
- ◆ 分析排序算法,如冒泡排序、插入排序、选择排序、合并排序和快速排序
- ◆ 使用大 O 符号检查算法复杂度和效率测量
- ◆ 绘制图形并执行深度 (DFS) 和宽度 (BFS) 遍历

模块4.Backend开发 III - Flask, 从零开始创建 API 和基本架构

- ◆ 使用 Flask 开发 RESTful API
- ◆ 将数据库集成到 Flask 应用程序中
- ◆ 在 API 中实现身份验证和安全性
- ◆ 使用 Flask 设计 backend 应用程序的基本架构

模块5.从零开始的面向对象编程和设计模式

- ◆ 定义面向对象编程的关键概念, 例如类、对象、属性、方法、封装、抽象、继承和多态性
- ◆ 通过使用 Python 编程语言了解类中封装和抽象的使用
- ◆ 检查 Python 语言中的多态性和重载的概念, 了解它们的应用和优势
- ◆ 确定类之间的关系类型, 例如关联、聚合和组合

模块6.Frontend I - 从头开始学习 HTML 和 CSS

- ◆ 识别 HTML 文档的基本结构及其在 Web 开发中的重要性
- ◆ 使用 HTML 以语义化和可访问的方式组织和呈现 Web 内容: Web 架构
- ◆ 使用 CSS 样式来改善元素的视觉呈现
- ◆ 使用 CSS 盒模型来构造和布局界面中的元素



模块7. Frontend II – 从零开始学习 JavaScript

- ◆ 理解 JavaScript 语法和数据类型
- ◆ 学习使用函数和控制结构来构造代码
- ◆ 操作 DOM 与动态网页进行交互
- ◆ 使用 Promises 和 async/await 处理 API 和异步

模块8. Frontend III - 从头开始使用 React.js

- ◆ 了解如何使用 JSX 创建声明式接口
- ◆ 学习使用功能组件、道具和生命周期
- ◆ 使用 Context API 和 Redux Toolkit 等现代工具管理本地和全局状态
- ◆ 实现路由以构建单页应用程序 (SPA)

模块9. 从零开始的数据库管理与优化

- ◆ 认识数据库的类型及其特征
- ◆ 理解并应用关系数据模型
- ◆ 培养数据库管理的 SQL 技能
- ◆ 使用 SQL 中的高级查询

模块10. 从头开始的开发工具：Linux、版本控制、CI/CD、Docker 和敏捷方法

- ◆ 在命令行级别管理 Linux 操作系统
- ◆ 掌握 Git 版本控制
- ◆ 实施持续集成和部署 (CI/CD) 管道
- ◆ 创建和管理 Docker 容器



你想掌握未来的语言吗?这个校级硕士学位代表着迈向你一直渴望的技术未来的第一步。现在报名吧!”

05 职业前景

这个学位在不断扩大的技术领域开辟了广泛的职业机会。随着各行各业的公司都走向数字化,对具有编程技能的专业人员的需求不仅在增加,而且在多样化,在软件开发、数据库管理、网页设计和流程自动化等各个领域开辟了机会。毕业后,毕业生将完全有资格在科技公司、初创企业和任何行业的组织中担任角色,这些公司希望将数字化解决方案融入其运营。





“

你不仅将成为一名顶级专家, 还将获得各种各样的职业机会, 确保你在不断增长和转型的就业市场中取得成功”

毕业生简介

毕业生将是一个训练有素的专业人士，以面对未来的技术挑战。在这方面，您不仅将掌握算法、数据结构和编程语言的基本概念，而且还将对行业中最需要的工具和技术有深入的了解。凭借在编程和应用程序开发方面的坚实基础，这位专家将能够自主有效地处理技术项目。此外，您还将因解决复杂问题、应用创新解决方案以及管理系统和数据库的能力而脱颖而出。

您将成为一名多才多艺的专业人士，拥有扎实的技术准备和战略眼光，能够将编程整合到响应公司和社会的技术解决方案中。

- ◆ **批判性思维和解决问题:**通过编程分析复杂情况并找到创新和高效的解决方案
- ◆ **团队合作和多学科协作:**在 multidisciplinary 团队中工作，与来自不同领域的专业人士如设计师、工程师和数据分析师有效合作，以实现技术项目的共同目标
- ◆ **时间管理和学习自主性:**有效地管理时间，以应对紧迫的最后期限，并独立负责地开发项目
- ◆ **适应新技术:**管理新的工具和编程语言，确保它们在快速发展的技术环境中不断更新



完成校级学位课程后,您将能够在以下职位上运用您的知识和技能:

1. **软件开发员:**负责根据客户或公司的需求设计、创建和维护应用程序和计算机程序。
2. **网络开发员:**管理网站的创建、设计和维护,确保其不同设备上的功能和优化。
3. **数据库工程师:**负责设计、实施和管理数据库,确保其效率、安全性和可用性。
4. **系统分析师:**现有 IT 系统的开发人员,提出改进建议以优化流程的性能和效率。
5. **Front -End程序员:**负责开发应用程序和网站的视觉部分,改善用户体验。
6. **Back-End程序员:**负责开发应用程序和网站的服务器逻辑和数据库,确保其正常运行。
7. **网络管理员:**负责管理和维护公司内部的网络基础设施,确保其稳定性和安全性。
8. **技术顾问:**公司的指导者和顾问,提供有关实施适合其需求的技术解决方案的信息,提高其绩效和竞争力。
9. **移动应用程序开发人员:**移动设备应用程序的设计、开发和维护经理,确保其可用性和性能。
10. **自动化软件工程师:**负责设计和开发自动化解决方案,以提高业务流程的效率,使用编程和先进技术。



您将有一个完整的内容在 100%在线模式提供一天 24 小时。加入 TECH 并享受它为您带来的所有好处!”

06 学习方法

TECH 是世界上第一所将案例研究方法与 Relearning 一种基于指导性重复的100% 在线学习系统相结合的大学。

这种颠覆性的教学策略旨在为专业人员提供机会, 以强化和严格的方式更新知识和发展技能。这种学习模式将学生置于学习过程的中心, 让他们发挥主导作用, 适应他们的需求, 摒弃传统方法。



“

我们的课程使你准备好在不确定的环境中面对新的挑战并获得事业上的成功”

学生:所有TECH课程的首要任务

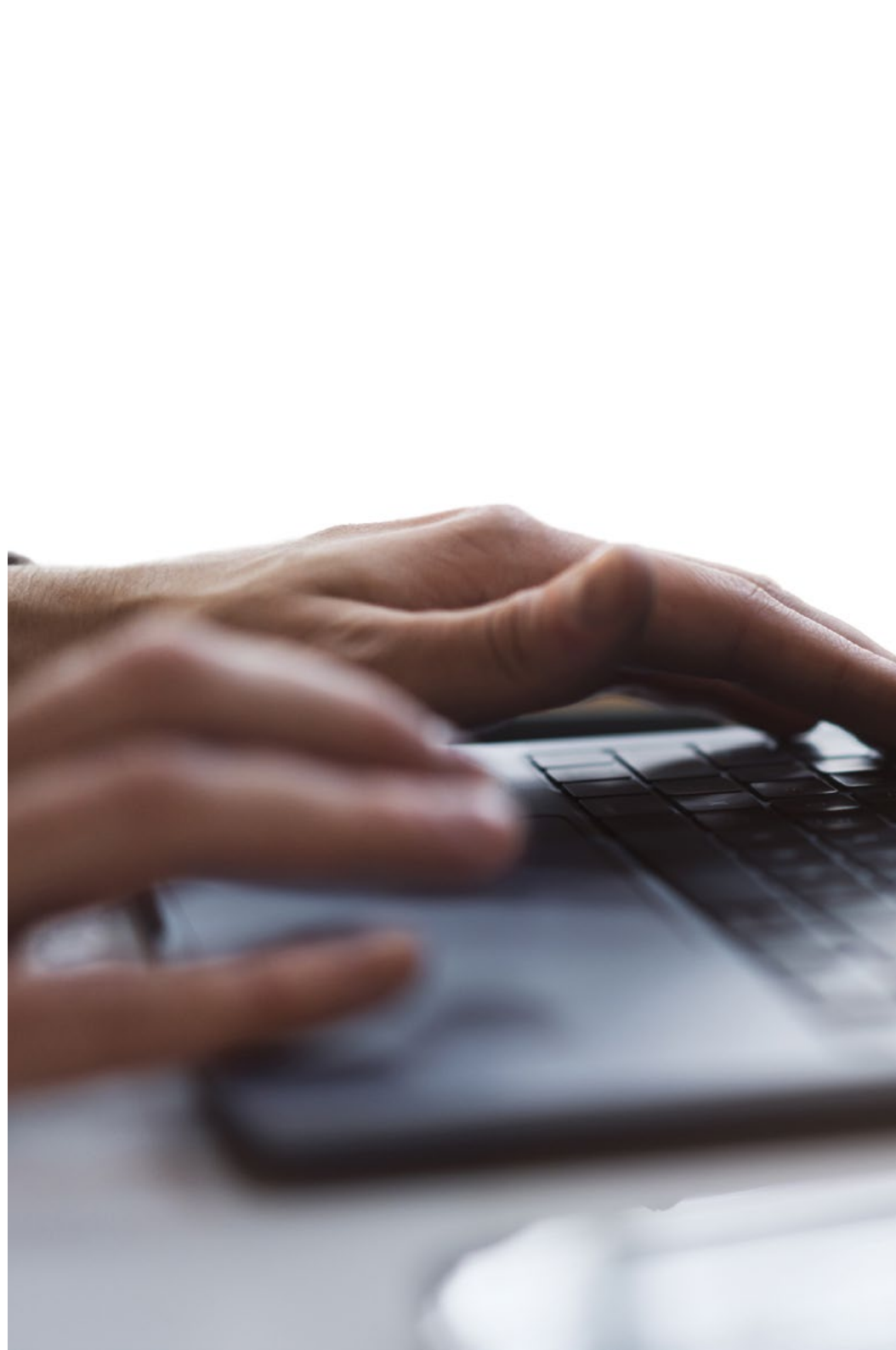
在 TECH 的学习方法中, 学生是绝对的主角。

每个课程的教学工具的选择都考虑到了时间, 可用性和学术严谨性的要求, 这些要求如今不仅是学生的要求也是市场上最具竞争力的职位的要求。

通过TECH的异步教育模式, 学生可以选择分配学习的时间, 决定如何建立自己的日常生活以及所有这一切, 而这一切都可以在他们选择的电子设备上舒适地进行。学生不需要参加现场课程, 而他们很多时候都不能参加。您将在适合您的时候进行学习。您始终可以决定何时何地学习。

“

在TECH, 你不会有线下课程(那些你永远不能参加)”



国际上最全面的学习计划

TECH的特点是提供大学环境中完整的学术大纲。这种全面性是通过创建教学大纲来实现的，教学大纲不仅包括基本知识，还包括每个领域的最新创新。

通过不断更新，这些课程使学生能够跟上市场变化并获得雇主最看重的技能。通过这种方式，那些在TECH完成学业的人可以获得全面的准备，为他们的职业发展提供显著的竞争优势。

更重要的是，他们可以通过任何设备，个人电脑，平板电脑或智能手机来完成的。

“

TECH模型是异步的，因此将您随时随地使用PC，平板电脑或智能手机学习，学习时间不限”

案例研究或案例方法

案例法一直是世界上最好的院系最广泛使用的学习系统。该课程于1912年开发，目的是让法学专业学生不仅能在理论内容的基础上学习法律，还能向他们展示复杂的现实生活情境。因此，他们可以做出决策并就如何解决问题做出明智的价值判断。1924年被确立为哈佛大学的一种标准教学方法。

在这种教学模式下，学生自己可以通过耶鲁大学或斯坦福大学等其他知名机构使用的边做边学或设计思维等策略来建立自己的专业能力。

这种以行动为导向的方法将应用于学生在TECH进行的整个学术大纲。这样你将面临多种真实情况，必须整合知识，调查，论证和捍卫你的想法和决定。这一切的前提是回答他在日常工作中面对复杂的特定事件时如何定位自己的问题。



学习方法

在TECH, 案例研究通过最好的100%在线教学方法得到加强: Relearning。

这种方法打破了传统的教学技术, 将学生置于等式的中心, 为他们提供不同格式的最佳内容。通过这种方式, 您可以回顾和重申每个主题的关键概念并学习将它们应用到实际环境中。

沿着这些思路, 根据多项科学研究, 重复是最好的学习方式。因此, TECH在同一课程中以不同的方式重复每个关键概念8到16次, 目的是确保在学习过程中充分巩固知识。

Relearning 将使你的学习事半功倍, 让你更多地参与到专业学习中, 培养批判精神, 捍卫论点, 对比观点: 这是通往成功的直接等式。



100%在线虚拟校园,拥有最好的教学材料

为了有效地应用其方法论,TECH 专注于为毕业生提供不同格式的教材:文本,互动视频,插图和知识图谱等。这些课程均由合格的教师设计,他们的工作重点是通过模拟将真实案例与复杂情况的解决结合起来,研究应用于每个职业生涯的背景并通过音频,演示,动画,图像等基于重复的学习。

神经科学领域的最新科学证据表明,在开始新的学习之前考虑访问内容的地点和背景非常重要。能够以个性化的方式调整这些变量可以帮助人们记住知识并将其存储在海马体中,以长期保留它。这是一种称为神经认知情境依赖电子学习的模型,有意识地应用于该大学学位。

另一方面,也是为了尽可能促进指导者与被指导者之间的联系,提供了多种实时和延迟交流的可能性(内部信息,论坛,电话服务,与技术秘书处的电子邮件联系,聊天和视频会议)。

同样,这个非常完整的虚拟校园将TECH学生根据个人时间或工作任务安排学习时间。通过这种方式,您将根据您加速的专业更新,对学术内容及其教学工具进行全局控制。



该课程的在线学习模式将您安排您的时间和学习进度,使其适应您的日程安排”

这个方法的有效性由四个关键成果来证明:

1. 遵循这种方法的学生不仅实现了对概念的吸收,而且还通过练习评估真实情况 and 应用知识来发展自己的心理能力。
2. 学习扎根于实践技能使学生能够更好地融入现实世界。
3. 由于使用了现实中出现的情况,思想和概念的学习变得更加容易和有效。
4. 感受到努力的成效对学生是一种重要的激励,这会转化为对学习更大的兴趣并增加学习时间。

最受学生重视的大学方法

这种创新学术模式的成果可以从TECH毕业生的整体满意度中看出。

学生们对教学质量,教材质量,课程结构及其目标的评价非常好。难怪该机构成为根据global score评分被学生评为最受欢迎的大学,获得了5分中的4.9分。

由于TECH掌握着最新的技术和教学前沿,因此可以从任何具有互联网连接的设备(计算机,平板电脑,智能手机)访问学习内容。

你可以利用模拟学习环境和观察学习法(即向专家学习)的优势进行学习。



因此,在这门课程中,将提供精心准备的最好的教育材料:



学习材料

所有的教学内容都是由教授这门课程的专家专门为这门课程创作的,因此,教学的发展是具体的。

这些内容之后被应用于视听格式,这将创造我们的在线工作方式,采用最新的技术,使我们能够保证给你提供的每一件作品都有高质量。



技能和能力的实践

你将开展活动以发展每个学科领域的具体能力和技能。在我们所处的全球化框架内我们提供实践和氛围帮你获得成为专家所需的技能和能力。



互动式总结

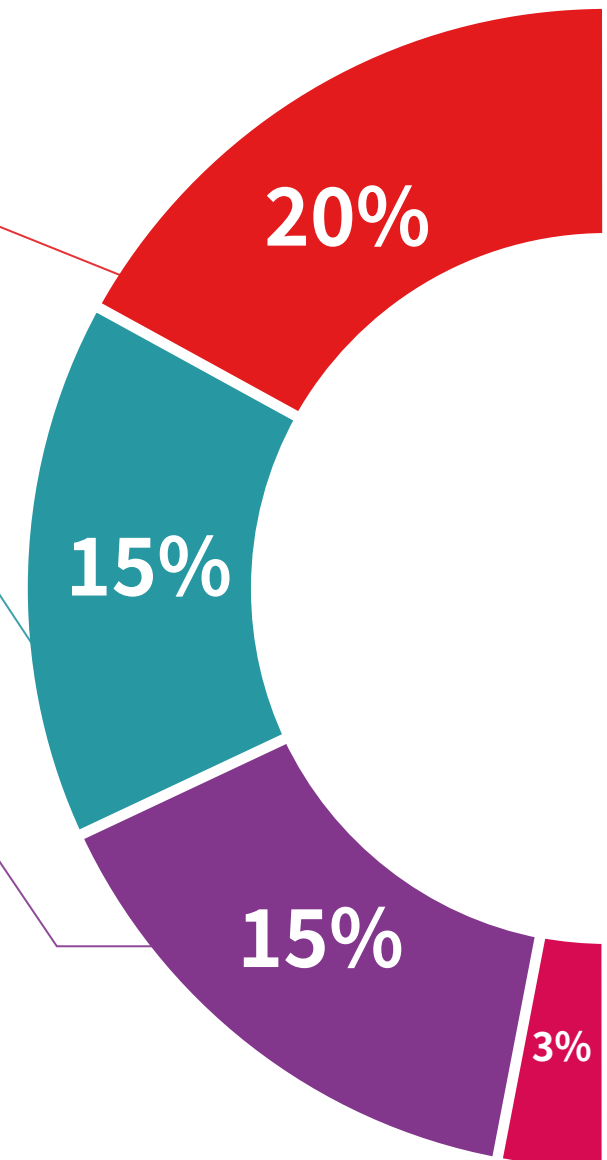
我们以有吸引力和动态的方式将内容呈现在多媒体中,包括音频,视频,图像,图表和概念图,以巩固知识。

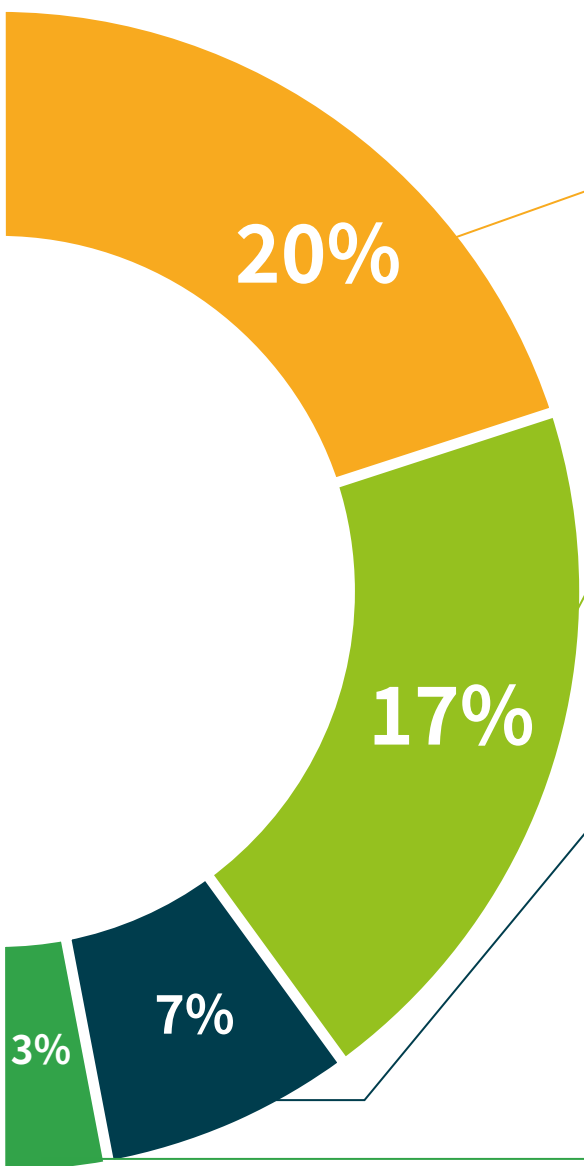
这一用于展示多媒体内容的独特教育系统被微软公司评为"欧洲成功案例"。



延伸阅读

最新文章,共识文件,国际指南...在我们的虚拟图书馆中,您将可以访问完成培训所需的一切。





案例研究

您将完成一系列有关该主题的最佳案例研究。由国际上最优秀的专家介绍,分析和指导案例。



Testing & Retesting

在整个课程中,我们会定期评估和重新评估你的知识。我们在米勒金字塔的4个层次中的3个层次上这样做。



大师班

科学证据表明第三方专家观察的效果显著。向专家学习可以增强知识和记忆力,并为我们今后做出艰难的决定建立信心。



快速行动指南

TECH以工作表或快速行动指南的形式提供课程中最相关的内容。一种帮助学生在学习中进步的综合,实用和有效的方法。



07

教学人员

教学团队由在技术领域具有丰富经验的高素质专业人员组成。每一个都是精心挑选的，以确保学生接受高质量的培训，基于最新的知识和实际的行业实践。从这个意义上说，他在知名公司的工作经验和他的学术准备使他能够提供一个全面的和应用的角度的编程。简而言之，导师是不同技术领域的专家，提供了多学科和丰富的见解。



“

教师队伍是该课程成功的基本支柱。导师的素质和经验将确保全面的培训，这将使你准备好面对技术部门的信心和能力”

管理



Lucas Cuesta, Juan Manuel 博士

- ◆ Indizen 高级软件工程师兼分析师 - 相信人才
- ◆ Krell Consulting 和 IMAGiNA 人工智能的高级软件工程师和分析师
- ◆ 英特尔公司软件工程师
- ◆ 智能对话系统公司软件工程师
- ◆ 他是马德里理工大学智能环境电子系统工程专业的博士
- ◆ 毕业于马德里理工大学电信工程专业
- ◆ 马德里理工大学智能环境系统电子工程硕士



Márquez Ruiz de Lacanal, Juan Antonio 先生

- ◆ GTD Defense & Security Solutions 的软件开发人员
- ◆ Solera Inc 的软件开发人员
- ◆ GRVC 塞维利亚研发工程师
- ◆ Unmute 联合创始人
- ◆ VR Educa 联合创始人
- ◆ 加州大学伯克利分校工程与创业学术交流
- ◆ 塞维利亚大学工业工程学位

教师

Luna Perejón, Francisco博士

- ◆ 计算机架构与技术专家
- ◆ 塞维利亚大学计算机工程博士学位
- ◆ 维哥大学计算机工程硕士学位
- ◆ 塞维利亚大学卫生工程学位
- ◆ 塞维利亚大学计算机工程与信息技术学位
- ◆ 成员: 机器人与信息技术研究小组 (TEP108)

Péris Millán, Eduardo先生

- ◆ 技术咨询领域总监
- ◆ 计算机工程专家
- ◆ 组织信息与知识战略管理硕士学位
- ◆ 领导力与公共管理硕士
- ◆ 公共管理专家
- ◆ 智慧城市计算系统专家

Pi Morell, Oriol 先生

- ◆ Fihoca 功能分析师
- ◆ CDmon 托管和电子邮件产品负责人
- ◆ Atmira 和 CapGemini 的功能分析师和软件工程师
- ◆ Capgemini 讲师, 组建 Capgemini 和 Atmira
- ◆ 巴塞罗那自治大学计算机管理技术工程学位
- ◆ 阿维拉天主教大学的人工智能硕士
- ◆ IMF Smart Education 工商管理硕士学位
- ◆ IMF Smart Education 信息系统管理硕士学位
- ◆ 加泰罗尼亚大学 (UOC) 设计模式研究生课程

Grillo Hernández, José Enrique博士

- ◆ 应用程序开发人员和技术分析师
- ◆ Globant高级移动应用程序开发人员
- ◆ Plexus Tech的Android开发人员
- ◆ RoadStr的高级Android开发人员
- ◆ AvantgardelT-信息技术服务高级移动开发人员
- ◆ Smartdess 项目负责人
- ◆ 在教育平板电脑中开发
- ◆ 企业移动解决方案技术分析师
- ◆ 西蒙-玻利瓦尔大学的系统工程硕士学位

Domínguez Valderrama, Desirée女士

- ◆ Lead Product & Growth Strategist
- ◆ 塞维利亚商会商学院平面设计与创意硕士学位
- ◆ CoderHouse 的UX/UI 设计师专家
- ◆ 科技与创业专家



趁此了解这个领域的最新发展并将其应用到你的日常工作中的机会”

08 学位

从零开始编程校级硕士除了保证最严格和最新的培训外,还可以获得由 TECH Global University 颁发的校级硕士学位证书。





“

顺利完成该课程后你将获得大学学位证书无需出门或办理其他手续”

这个课程将使您有机会获得 **TECH Global University** 认可的**从零开始编程校级硕士学位**。**TECH Global University** 是全球最大的数字大学。

TECH Global University 是一所经安道尔政府 ([官方公报](#)) 公开认可的欧洲官方大学。自2003年以来,安道尔已成为欧洲高等教育区 (EEES) 的一部分。该高等教育区是欧盟推动的一个倡议,旨在组织国际教育框架,并协调成员国的高等教育系统。该项目促进了共同价值观的推广,实施了共同工具,并加强了质量保证机制,以促进学生、研究人员和学者之间的合作和流动。

TECH Global University 的专业学位是一个欧洲的继续教育和职业更新项目,确保学生在其知识领域获得能力,并为完成该项目的学生赋予了高度的学术价值。

学位:**从零开始编程校级硕士**

模式:**在线**

时长:**12个月**

认证:**60 ECTS 学分**



*海牙使馆认证。如果学生要求其纸质学位证书获得海牙使馆认证,TECH Global University 将为其进行相关手续,但需支付额外费用。

健康 信心 未来 人 导师
教育 信息 教学
保证 资格认证 学习
机构 社区 科技 承诺
个性化的关注 现在
知识 网页 培
网上教室 发展 语言

tech global
university

校级硕士 从零开始编程

- » 模式: 在线
- » 时长: 12个月
- » 学位: TECH Global University
- » 认证: 60 ECTS 学分
- » 课程表: 自由安排时间
- » 考试模式: 在线

校级 硕士

从零开始编程

