校级硕士
Python开发

tech global university

校级硕士
Python开发

- 模式:**在线**
- 时长: **7个月**
- 学位: **TECH Global University**
- 认证: **ECTS 60**
- 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/professional-master-degree/master-python-development

目录

01

介绍

Python是一种多功能且高效的编程语言。首先，其清晰易读的语法使编写和理解代码变得容易，从而加快了开发过程并减少了出错的可能性。此外，丰富多样的库和框架为开发者提供了强大而现成的工具，允许更快速和高效的开发。其开源性质和活跃的开发者社区也有助于该语言的不断改进和更新，保证了动态和现代的开发环境。为此，TECH 开发了一门详尽的课程旨在通过100%在线方法培训专业人员掌握Python开发的最新进展。

Python以其适应性而闻名，被广泛用于从Web开发到人工智能的应用程序中。你还在等什么呢?现在就报名吧"

Python是一种高级编程语言，在计算机科学家中广泛使用，因为它拥有广泛的库和框架，可以简化常见任务使开发人员能够专注于应用程序的逻辑而不是将时间浪费在技术细节上。此外，它的多功能性是它的另一个显着优势，因为它可以在各种环境中使用，从网络开发到数据分析和机器学习。

这就是Python开发硕士学位的诞生，该课程将包括Python中的高级数据和类型管理，探索标识符，关键字，整数和布尔类型以及高级字符串格式和编码。此外，还将检查元组，列表和字典等集合，以及迭代技术和 lambda函数，为该语言的基础知识奠定坚实的基础。

同样，将深入研究Python应用程序的开发，强调最佳实践和现代方法。从应用程序架构到部署和维护，将包括高级设计和建模,依赖管理,设计模式,测试和调试,性能优化以及部署和分发策略等方面。

同样，计算机专业人士将深入研究使用Python进行网页和移动应用开发包括Django和Flask等框架，以及API和网络服务的开发。此外，您将专注于界面和用户体验 (UI/UX) 设计，从设计工具的使用到提高可访问性和可用性。最后，将使用Python和 NumPy,Pandas和Matplotlib等工具进行数据管理和分析。

因此，通过完全在线且适应性强的学术提案，该学位被视为一个独特的机会。通过这种方法，专业人士将享有更大的自由来管理他们的访问时间，从而使他们能够协调日常的个人和工作承诺。

这个**Python开发校级硕士**包含市场上最完整和最新的课程。主要特点是：

- Python开发专家呈现的开发实战案例
- 课程包括图形化，示意图和实用性内容提供了关于那些对专业实践至关重要的学科的理论和实践信息
- 进行自我评估以改善学习的实践练习
- 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
- 可以通过任何连接互联网的固定或便携设备访问课程内容

> Python 开发方面的全面且专业的培训，将使您做好准备面对软件开发领域的真正挑战”

通过这门100%在线硕士学位，您将深入研究应用程序商店中的前端和后端设计，数据库管理和发布策略”

您将包括从变量和数据类型等基础知识到最先进的数据可视化技术和性能优化策略的所有内容和存储。现在就报名吧!。

您将深入研究面向对象编程(OOP)以及类，继承，多态性，抽象类的创建和自定义异常等主题。

这门课程的教学人员包括来自这个行业的专业人士，他们将自己的工作经验带到了这一培训中还有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。

这门课程的设计集中于基于问题的学习，通过这种方式专业人士需要在整个学年中解决所遇到的各种实践问题。为此，你将得到由知名专家制作的新型交互式视频系统的帮助。

02

目标

该硕士学位的设计宗旨是在这种高性能语言的广阔编程领域培养高素质和专业的专业人员。通过详尽的模块，毕业生将深入了解Python的语法和功能，从高级数据管理到掌握面向对象编程(OOP)以及Web和移动应用程序的高效设计。该课程采用严格且最新的教学方法，将保证独特的教育体验使计算机科学家成为软件开发行业令人垂涎的专家。准备好突破自我，踏上一段教育之旅不仅提升您的Python技能还为您提供在动态科技发展领域中脱颖而出和领导的工具。

成为软件开发领域的
领导者，为他们配备先
进的知识和专业技能
以应对当代技术挑战”

总体目标

- 提供对Python的全面理解
- 使用Python训练高级数据和类型管理
- 在Python中应用面向对象编程 (OOP)的原理
- 促进在软件开发中使用最佳实践和现代方法
- 提供使用Python进行Web和移动开发的全面培训
- 将UI/UX原则集成到软件开发中
- 数据开发工具和环境的配置和使用培训
- 深入研究Python中数据结构和函数的使用
- 使用Matplotlib培训高级数据可视化技术
- 数据存储和性能优化策略培训

> 得益于该学位提供的内容丰富的虚拟图书馆，其中充满了最具创新性的多媒体资源，您将获得在动态的技术发展世界中脱颖而出并处于领先地位的工具”

具体目标

模块 1. 用Python编程
- 启用Python开发环境的配置和有效使用
- 了解高级编程概念

模块 2. 使用Python进行高级数据和流控制
- 掌握管理标识符和关键字的约定和实践
- 应用复杂的数据结构及其操作

模块 3. Python中的面向对象编程(OOP)
- 掌握Python中类和对象的创建和使用
- 在Python中应用继承和多态

模块 4. Python应用程序开发
- 专注于高级应用程序设计和建模
- 了解应用优化，部署和维护

模块 5. 使用Python进行Web和移动开发
- 在Python中使用流行的Web 框架
- 准备移动应用程序开发和发布

模块 6. Python的界面和用户体验
- 教授响应式和自适应设计技术
- 准备执行可用性测试和用户行为分析

模块 7. 使用Python进行数据处理和大数据
- 处理数据管理的流控制技术和功能
- 推广Python编码和错误处理最佳实践

模块 8. Python中的数据结构和函数
- 以高级方式创建和使用函数
- 读写文件及其处理

模块 9. 使用NumPy和Pandas在Python中进行数据管理
- 使用NumPy创建和操作数组
- 使用Matplotlib促进数据可视化领域的竞争

模块 10. NumPy和Pandas的先进技术和实际应用
- 培养从各种来源加载和存储数据的专业知识
- 指导高级数据清理和转换策略

03

能力

该课程不仅传授技术知识，还将注重培养技能，将毕业生转变为软件开发领域的杰出专业人员。因此，通过对Python的专家掌握，将包括从高级数据管理到创建高性能Web和移动应用程序的开发创新和高效解决方案的能力。此外，计算机科学家将具备面向对象编程（OOP），用户界面和用户体验（UI/UX）设计以及使用NumPy和Pandas等库进行高级数据分析的专业技能。

将鼓励关键软技能的发展，例如批判性思维，解决问题，团队协作和适应能力”

总体能力

- 培养实用的编程技能
- 为自己配备程序流程控制方面的高级技能
- 处理面向对象软件的设计和实现
- 培训Python应用程序的综合开发
- 掌握网络和移动应用程序的设计和管理
- 使用Python进行控制界面和用户体验设计
- 使用Python培养数据管理和分析技能
- 获得高级文件管理和Python建模技能
- 使用NumPy和Pandas培养高级数据管理技能
- 使用NumPy和Pandas深入研究高级数据管理

> 您将面临现实世界的挑战并在竞争激烈且不断变化的就业市场中脱颖而出。选择TECH吧”

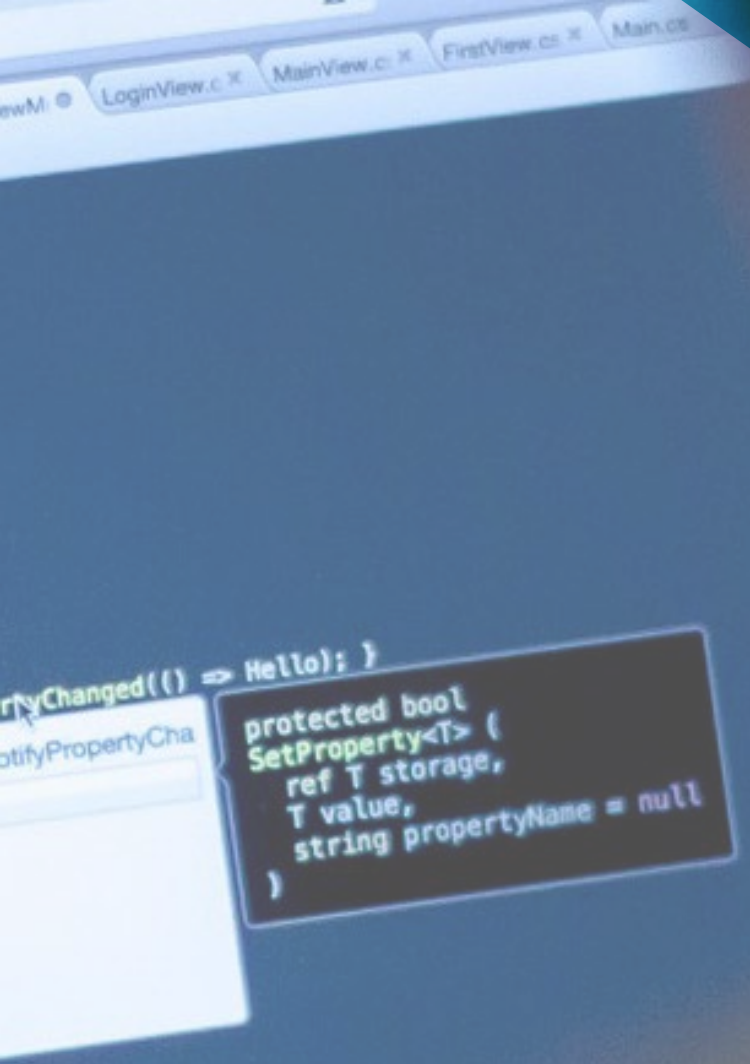
具体能力

- Python高级数据管理培训
- 处理Python中函数的高级使用
- 实现高级面向对象编程 (OOP) 概念，例如抽象类和自定义异常
- 控制测试和调试
- 开发API和Web服务
- 掌握PythonUI/UX设计工具的使用
- 在Python中使用数据的基本库
- 以实际的方式应用不同的数据结构
- 使用Pandas进行结构化数据管理
- 提升时间序列分析和操作的技能和复杂的数据

04

课程管理

该硕士学位的教师都是行业专家，在软件开发方面拥有丰富的实践经验和成功项目的良好记录。这些导师将理论与实际应用联系起来，不仅传授Python的高级知识，还会分享他们的实际经验，为学生提供有关行业当前最佳实践和趋势的宝贵见解。此外，其教学方法将因其致力于为毕业生提供最新的技术培训而脱颖而出，以在动态的软件开发领域脱颖而出。



> 您将与最优秀的人一起获得成功，我们将获得您进入Python开发领域所需的知识和技能”

管理人员

Matos Rodríguez, Dionis 先生

- Wide Agency Sodexo数据工程师
- Tokiota数据顾问
- Devoteam数据工程师
- Ibermática的BI开发人员
- Johnson Controls应用工程师
- Suncapital Spain的数据库开发人员
- Deadlock Solutions的高级Web开发人员
- Metaconxept的QA分析师
- EAE 商学院大数据与分析硕士
- 系统分析与设计硕士
- APEC大学计算机工程学士学位

教师

Villar Valor, Javier 先生

- Impulsa2董事兼创始合伙人
- Summa Insurance Brokers 首席运营官 (COO)
- Johnson Controls转型与卓越运营总监
- 专业教练硕士
- 法国里昂商学院EMBA
- EOI质量管理硕士
- Acción Pro-Educación y Cultura大学 (UNAPEC) 计算机工程

Gil Contreras, Armando 先生

- Jhonson Controls 首席大数据科学家
- Opensistemas S.A.大数据科学家
- Creatividad y Tecnología S.A. 基金审计员 (中国交通运输协会)
- 普华永道会计师事务所公共部门审计师
- 大学技术与艺术中心数据科学硕士
- (CEF 金融研究中心国际关系与商业 MBA)
- 圣多明各理工学院经济学学士学位

Gil Contreras, Milagros 女士

- MPCTech LLC的 内容创作者
- 专案经理
- 自由职业IT作家
- 马德里康普顿斯大学 MBA
- 该学院工商管理专业毕业生圣多明各理工学院

Delgado Feliz, Benedit 女士

- 国家禁毒总局行政助理兼电子监控操作员
- 卡塞雷斯和设备的客户服务
- Express Parcel Services (EPS) 的索赔和客户服务
- 国立信息学院Microsoft Office专家
- 圣多明各天主教大学社会沟通师

> 借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"

05

结构和内容

课程大纲经过精心设计旨在为毕业生提供全面的体验。从深入研究Python语法和高级功能，到面向对象编程 (OO P)，Web和移动应用程序设计以及使用NumPy和Pandas 等库进行专家数据管理，每个模块都经过精心构建以提供扎实的知识和实践技能。此外，还将探讨关键主题，例如用户界面和用户体验 (UI/UX) 设计，高级数据管理以及性能和存储优化。

> 您不仅将完全熟练掌握Python，而且还将做好准备，自信地面对技术行业的动态和挑战”

模块 1. 用Python编程

1.1. 在Python中创建和运行程序
- 1.1.1. 开发环境配置
- 1.1.2. 在Python中运行脚本
- 1.1.3. 集成开发工具 (IDE)

1.2. Python中的数据
- 1.2.1. 基本类型 (int, float, str)
- 1.2.2. 在Python中转换和转换数据类型
- 1.2.3. Python中的不变性和数据存储

1.3. Python中的对象引用
- 1.3.1. 内存中的引用
- 1.3.2. 身份对平等
- 1.3.3. 引用管理和垃圾收集

1.4. 用Python收集数据
- 1.4.1. 列表和常用操作
- 1.4.2. 元组及其不变性
- 1.4.3. 字典和数据访问

1.5. Python中的逻辑运算
- 1.5.1. 布尔运算符
- 1.5.2. 条件表达式
- 1.5.3. 短路评估

1.6. Python中的算术运算符
- 1.6.1. Python中的算术运算
- 1.6.2. 分部经营者
- 1.6.3. 优先级和关联性

1.7. Python中的输入/输出
- 1.7.1. 从标准输入读取数据
- 1.7.2. 将数据写入标准输出
- 1.7.3. 文件管理

1.8. 在Python中创建和调用函数
- 1.8.1. 函数语法
- 1.8.2. 参数和参数
- 1.8.3. 返回值和匿名函数

1.9. 在Python中使用字符串
  1.9.1. 字符串操作和格式化
  1.9.2. 常用字符串方法
  1.9.3. 插值和F弦
1.10. Python中的错误和异常处理
  1.10.1. 常见的异常类型
  1.10.2. 块try-except
  1.10.3. 创建自定义异常

模块 2 .使用Python进行高级数据和流控制

2.1. Python中的标识符和关键字
  2.1.1. 变量名称规则
  2.1.2. Python中的保留字
  2.1.3. 命名约定
2.2. Python中的整型和布尔类型
  2.2.1. 综合类型
  2.2.2. 布尔特定运算
  2.2.3. 转换和表示
2.3. Python中的浮点类型和复数
  2.3.1. 准确性和代表性
  2.3.2. 浮点运算
  2.3.3. 在计算中使用复数
2.4. Python中的字符串格式和编码
  2.4.1. 高级格式化方法
  2.4.2. Unicode和 UTF-8 编码
  2.4.3. 使用特殊字符
2.5. 收藏：Python中的元组，列表和字典
  2.5.1. 类型之间的比较和对比
  2.5.2. 特定类型的方法
  2.5.3. 效率和正确的选型
2.6. Python中的集合和冻结集
  2.6.1. 集合的创建和操作
  2.6.2. 冻 结集
  2.6.3. 实际应用和性能
2.7. 在Python中迭代和复制集合
  2.7.1. For循环和列表推导式
  2.7.2. 浅拷贝对深度
  2.7.3. 迭代器和生成器
2.8. 在Python中使用LambdaLambda函数
  2.8.1. Lambda函数的语法和创建
  2.8.2. 在过滤器和地图中的应用
  2.8.3. 局限性和良好实践
2.9. 控制结构: Python中的条件和循环
  2.9.1. if-else和elif结构
  2.9.2. while和for循环
  2.9.3. 具有break, continue和else流控制
2.10. Python中的高级函数和方法
  2.10.1. 递归函数
  2.10.2. 高阶函数
  2.10.3. 函数装饰器

模块 3. Python中的面向对象编程(OOP)

3.1. Python中的面向对象编程(OOP)
  3.1.1. 类和对象
  3.1.2. 封装和抽象
  3.1.3. Python中的面向对象编程(OOP)
3.2. 在Python中创建类和对象
  3.2.1. Python中的OOP类
  3.2.2. 实例化和初始化方法
  3.2.3. 属性和方法
3.3. Python中的属性和方法
  3.3.1. 实例属性对层级
  3.3.2. 实例，类和静态方法
  3.3.3. 信息的封装和隐藏
3.4. Python中的继承和多态性
  3.4.1. 简单继承和多重继承
  3.4.2. 重写和扩展方法
  3.4.3. 多态性和Duck Typing

3.5. Python中的属性和属性访问
- 3.5.1. Getters和Setters
- 3.5.2. 装饰者@property
- 3.5.3. 访问控制和验证

3.6. Python中的自定义类和集合
- 3.6.1. 创建集合类型
- 3.6.2. 特殊方法（__len__, __getitem_,）
- 3.6.3. 自定义迭代器

3.7. Python类中的聚合和组合
- 3.7.1. 分类之间的关系
- 3.7.2. 聚合对组成
- 3.7.3. 对象生命周期管理

3.8. 在Python的类中使用装饰器
- 3.8.1. 方法的装饰器
- 3.8.2. 类装饰器
- 3.8.3. 应用程序和用例

3.9. Python中的抽象类和方法
- 3.9.1. 抽象类
- 3.9.2. 抽象方法及实现
- 3.9.3. 使用ABC（抽象基类）

3.10. Python OOP中的异常和错误处理
- 3.10.1. 类中的自定义异常
- 3.10.2. 处理方法中的异常
- 3.10.3. 异常和OOP的良好实践

模块 4. Python应用程序开发

4.1. Python中的应用程序架构
- 4.1.1. 软件设计
- 4.1.2. 常见的架构模式
- 4.1.3. 需求和需求评估

4.2. Python应用程序的设计和建模
- 4.2.1. 使用UML和图表
- 4.2.2. 数据建模和信息流
- 4.2.3. SOLID原则和模块化设计

4.3. Python中的依赖关系和库管理
- 4.3.1. 使用Pip处理包
- 4.3.2. 虚拟环境的使用
- 4.3.3. 解决依赖冲突

4.4. Python开发中的设计模式
- 4.4.1. 创造，结构和行为模式
- 4.4.2. 模式的实际应用
- 4.4.3. 重构和模式

4.5. Python应用程序中的测试和调试
- 4.5.1. 测试策略（单一，集成）
- 4.5.2. 测试框架的使用
- 4.5.3. 调试技术和工具

4.6. Python中的安全性和身份验证
- 4.6.1. 应用安全
- 4.6.2. 认证授权的实现
- 4.6.3. 漏洞防范

4.7. Python应用程序的优化和性能
- 4.7.1. 性能分析
- 4.7.2. 代码优化技巧
- 4.7.3. 资源和数据的高效管理

4.8. 使用Python部署和分发应用程序识别组织中的知识和人才　部署策略
- 4.8.2. 使用容器和编排器
- 4.8.3. 发行及持续更新

4.9. Python中的维护和更新
- 4.9.1. 软件生命周期管理
- 4.9.2. 维护和重构策略
- 4.9.3. 系统升级和迁移

4.10. Python文档和技术支持
- 4.10.1. 创建有效的文档
- 4.10.2. 文档工具
- 4.10.3. 与用户的支持和沟通策略

模块 5. 使用Python进行Web和移动开发

5.1. 使用Python进行Web开发
- 5.1.1. 网站的结构和组成部分
- 5.1.2. 网络开发技术
- 5.1.3. 网络开发趋势

5.2. 使用Python的流行Web框架
- 5.2.1. Django, Flask 和其他选项
- 5.2.2. 框架比较与选择
- 5.2.3. 与前端集成

5.3. 前端开发: 通过Python HTML, CSS和 JavaScript
- 5.3.1. HTML和CSS
- 5.3.2. JavaScript和DOM操作
- 5.3.3. 前端框架和库

5.4. 使用Python的后端和数据库
- 5.4.1. 使用Python进行后端开发
- 5.4.2. 关系型和非关系型数据库管理
- 5.4.3. 后端-前端集成

5.5. 使用Python的API和Web服务
- 5.5.1. RESTful API设计
- 5.5.2. API实施和文档
- 5.5.3. API中的消耗和安全性

5.6. 使用Python进行移动开发
- 5.6.1. 移动开发平台 (本机, 混合)
- 5.6.2. 开发工具和环境
- 5.6.3. 移动设备应用程序的适配

5.7. 使用Python的移动开发平台
- 5.7.1. 交叉开发框架
- 5.7.2. 移动设备上的测试和部署

5.8. 使用Python进行移动应用程序的设计和用户体验
- 5.8.1. 移动端界面设计
- 5.8.2. Python的界面和用户体验
- 5.8.3. 原型设计和设计工具

5.9. 使用Python在移动设备上进行测试和调试
- 5.9.1. 移动设备上的测试策略
- 5.9.2. 调试和监控工具
- 5.9.3. 自动化测试

5.10. 使用Python发布到应用商店
- 5.10.1. App Store和Google Play上的发布流程
- 5.10.2. 应用程序合规性和政策
- 5.10.3. 营销和促销策略

模块 6. Python的界面和用户体验

6.1. 使用Python进行用户界面设计
- 6.1.1. 使用Python进行UI设计
- 6.1.2. 使用Python进行人机交互
- 6.1.3. 使用Python以用户为中心的设计

6.2. 使用Python的 UI/UX设计工具
- 6.2.1. 设计和原型软件
- 6.2.2. 协作和反馈工具
- 6.2.3. 设计融入开发过程

6.3. 使用Python进行响应式和自适应设计
- 6.3.1. 响应式设计技术
- 6.3.2. 适应不同设备和屏幕
- 6.3.3. 测试和质量保证

6.4. 使用Python制作动画和过渡
- 6.4.1. 使用Python创建有效的动画
- 6.4.2. 动画工具和库
- 6.4.3. 对用户体验和性能的影响

6.5. Python的可访问性和可用性
- 6.5.1. 网站可访问性
- 6.5.2. 评估工具和技术
- 6.5.3. 实施最佳实践

6.6. 使用Python进行原型设计和线框图
- 6.6.1. 创建线框和模型
- 6.6.2. 快速原型制作工具
- 6.6.3. 可用性测试和反馈

6.7. 使用Python进行可用性测试
- 6.7.1. 可用性测试方法和技术
- 6.7.2. 根据结果进行分析和改进
- 6.7.3. 可用性测试工具

6.8. 使用Python进行用户行为分析
- 6.8.1. 分析和跟踪技术
- 6.8.2. 数据和指标的解释
- 6.8.3. 数据驱动的持续改进

6.9. 使用Python进行基于反馈的改进
- 6.9.1. 反馈管理与分析
- 6.9.2. 反馈周期和持续改进
- 6.9.3. 实施有效变革的策略

6.10. Python的 UI/UX未来趋势
- 6.10.1. 创新和新兴趋势
- 6.10.2. 新技术对UI/UX的影响
- 6.10.3. 为设计的未来做好准备

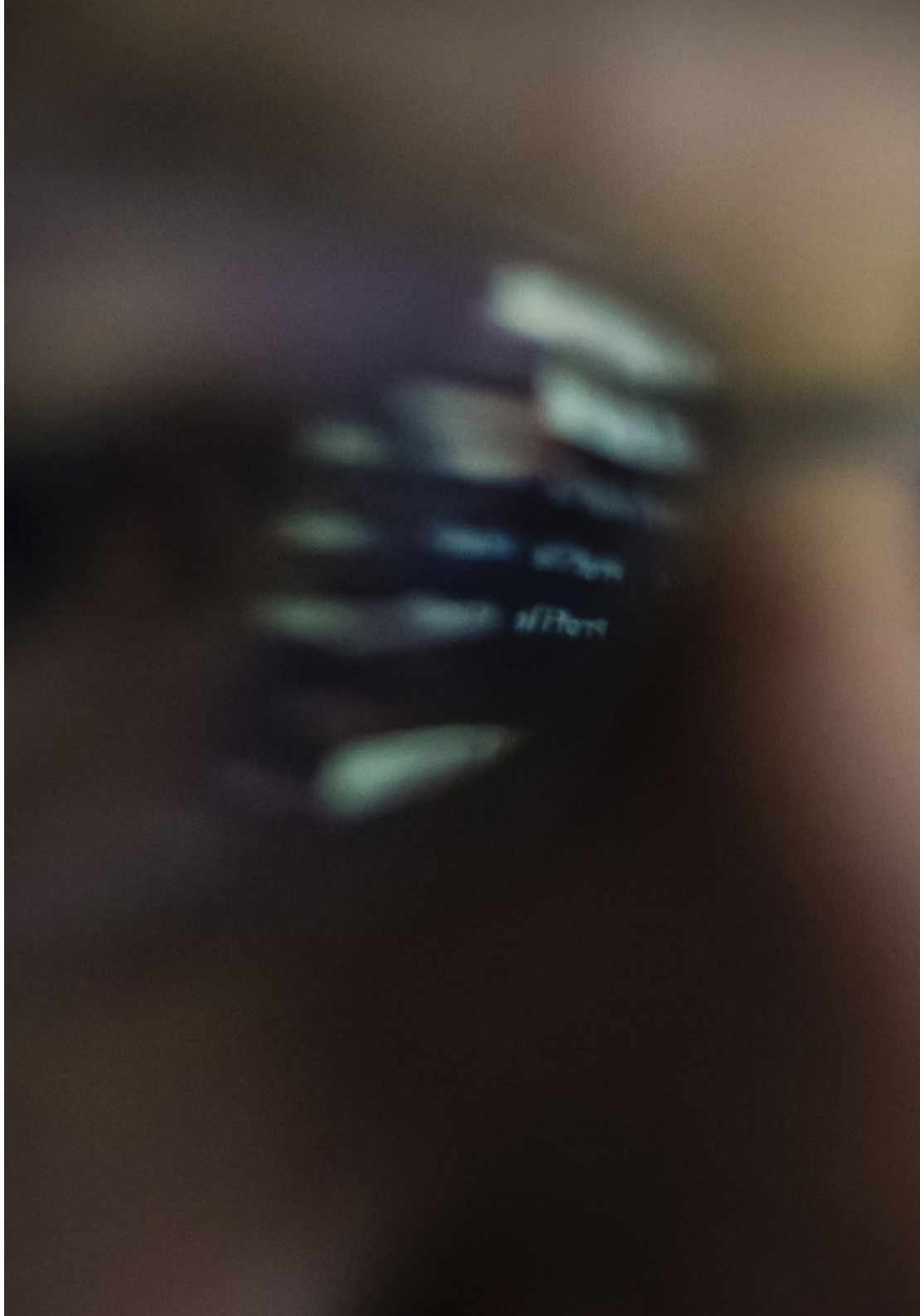
模块 7. 使用Python进行数据处理和大数据

7.1. 在数据上使用Python
- 7.1.1. Python在数据科学和分析中的应用
- 7.1.2. 基本数据库
- 7.1.3. 应用和实例

7.2. 使用Python搭建开发环境
- 7.2.1. Python安装和工具
- 7.2.2. 虚拟环境配置
- 7.2.3. 集成开发工具 (IDE)

7.3. Python中的变量，数据类型和运算符
- 7.3.1. 变量和原始数据类型
- 7.3.2. 数据结构
- 7.3.3. 算术和逻辑运算符

7.4. 流量控制：条件和循环
- 7.4.1. (if, else, elif)条件控制结构
- 7.4.2. 循环(for, while)和流程控制
- 7.4.3. 列表推导式和生成表达式



7.5. Python的函数和模块化
- 7.5.1. 使用函数
- 7.5.2. 参数，参数和返回值
- 7.5.3. 模块化和代码重用

7.6. 使用Python处理错误和异常
- 7.6.1.错误和异常
- 7.6.2. 使用try-except处理异常
- 7.6.3. 创建自定义异常

7.7. IPython工具
- 7.7.1. IPython工具
- 7.7.2. 使用IPython分析数据
- 7.7.3. 与标准Python解释器的差异

7.8. Jupyter Notebooks
- 7.8.1. Jupyter Notebooks
- 7.8.2. 使用笔记本分析数据
- 7.8.3. 发布Jupyter笔记本

7.9. Python最佳编码实践
- 7.9.1. 风格和约定 (PEP 8)
- 7.9.2. 文档和评论
- 7.9.3. 测试和调试策略

7.10. Python资源和社区
- 7.10.1. 在线资源和文档
- 7.10.2. 社区和论坛
- 7.10.3. Python的学习和更新

模块 8. Python中的数据结构和函数

8.1. Python中的集合
- 8.1.1. 操作与方法
- 8.1.2. 差异及实际应用
- 8.1.3. 迭代与理解

8.2. 字典及其在Python中的使用
- 8.2.1. 创建和操作字典
- 8.2.2. 数据访问和管理
- 8.2.3. 先进的模式和技术

8.3. Python中的列表和字典推导式
- 8.3.1. 语法和示例
- 8.3.2. 效率和可读性
- 8.3.3. 实际应用

8.4. Python中的数据函数
- 8.4.1. 特征创建
- 8.4.2. 范围和命名空间
- 8.4.3. 匿名函数和Lambda函数

8.5. Python中的函数参数和返回值
- 8.5.1. 位置参数和命名参数
- 8.5.2. 多个返回值
- 8.5.3. 可变参数和关键字

8.6. Python中的Lambda函数和高阶函数
- 8.6.1. 使用 Lambda函数
- 8.6.2. Map, Filter和 Reduce函数
- 8.6.3. 数据处理中的应用

8.7. Python中的文件处理
- 8.7.1. 读取和写入文件
- 8.7.2. 处理二进制和文本文件
- 8.7.3. 良好实践和异常处理

8.8. 在Python中读写文本和二进制文件
- 8.8.1. 文件格式和编码
- 8.8.2. 处理大文件
- 8.8.3. 序列化和反序列化(JSON, pickle)

8.9. 上下文和文件操作
- 8.9.1. 使用上下文管理器(with)
- 8.9.2. 文件处理技术
- 8.9.3. 安全和错误处理

8.10. Python中的建模库
- 8.10.1. Scikit-learn
- 8.10.2. TensorFlow
- 8.10.3. Pytorch

模块 9. 使用NumPy和Pandas在Python中进行数据管理

9.1. 在NumPy中创建和操作 数组
- 9.1.1. NumPy
- 9.1.2. 数组的基本操作
- 9.1.3. 数组的操作和转换

9.2. 数组的向量化运算
- 9.2.1. 矢量化
- 9.2.2. 通用函数(ufunc)
- 9.2.3. 效率和性能

9.3. NumPy中的索引和分段
- 9.3.1. 访问元素和Slicing
- 9.3.2. 高级和布尔索引
- 9.3.3. 重新排序和选择

9.4. Pandas series和DataFrames
- 9.4.1. Pandas
- 9.4.2. Pandas中的数据结构
- 9.4.3. DataFrames操作

9.5. Pandas中的索引和选择
- 9.5.1. 访问系列数据和DataFrames
- 9.5.2. 选择和过滤方法
- 9.5.3. loceiloc应用

9.6. 通过Pandas的操作
- 9.6.1. 算术运算和对齐
- 9.6.2. 聚合和统计功能
- 9.6.3. 函数的转换和应用

9.7. 处理Pandas中的不完整数据
- 9.7.1. 空值的检测和处理
- 9.7.2. 填充和删除不完整的数据
- 9.7.3. 处理不完整数据的策略

9.8. Pandas中的功能和应用
- 9.8.1. 串联和数据融合
- 9.8.2. 分组和聚合(groupby)
- 9.8.3. Pivot Tables 和Crosstabs

9.9. 使用Matplotlib进行可视化
- 9.9.1. Matplotlib
- 9.9.2. 图表创建和定制
- 9.9.3. 通过Pandas集成

9.10. 在Matplotlib中自定义图表
- 9.10.1. 样式和设置
- 9.10.2. 高级图形(scatter, bar, etc.)
- 9.10.3. 创建复杂的可视化

模块 10. NumPy和Pandas的先进技术和实际应用

10.1. 从不同来源加载数据
- 10.1.1. 从CSV, Excel和数据库导入
- 10.1.2. 从API和Web读取数据
- 10.1.3. 大数据管理策略

10.2. Python中的数据存储
- 10.2.1. 导出为不同格式
- 10.2.2. 存储效率
- 10.2.3. 数据安全和隐私

10.3. Python中的数据清理策略
- 10.3.1. 识别和纠正不一致之处
- 10.3.2. 数据标准化和转换
- 10.3.3. 清洁过程自动化

10.4. Pandas中的高级数据转换
- 10.4.1. 操纵和转换技术
- 10.4.2. 组合和重组DataFrames
- 10.4.3. 在Pandas中使用正则表达式

10.5. 在Pandas中组合Data Frames
- 10.5.1. Merge, Join 和串联
- 10.5.2. 冲突和密钥管理
- 10.5.3. 高效的组合策略

10.6. Pandas中数据的高级转换和旋转
- 10.6.1. Pivot和Melt
- 10.6.2. 重塑和转置技术
- 10.6.3. 数据分析中的应用

10.7. Pandas中的时间序列
- 10.7.1. 日期和时间管理
- 10.7.2. Resampling和Window Functions
- 10.7.3. 趋势和季节性分析

10.8. Pandas中的高级索引管理
- 10.8.1. 多级和分层索引
- 10.8.2. 高级选择和操作
- 10.8.3. 查询优化

10.9. 性能优化策略
- 10.9.1. 速度和效率的提高
- 10.9.2. 使用Cython和Numba
- 10.9.3. 并行化和分布式处理

10.10. 实用的数据处理项目
- 10.10.1. 开发实际使用示例
- 10.10.2. Python技术的集成
- 10.10.3. 解决复杂数据问题的策略

> 这门创新课程将适应您使您了解最新的趋势和技术，确保您处于软件开发创新的最前沿”

06

方法

这个培训计划提供了一种不同的学习方式。我们的方法是通过循环的学习模式发展起来的: **Re-learning**。

这个教学系统被世界上一些最著名的医学院所采用，并被**新英格兰医学杂志**等权威出版物认为是最有效的教学系统之一。

发现 Re-learning，这个系统放弃了传统的线性学习，带你体验循环教学系统：这种学习方式已经证明了其巨大的有效性，尤其是在需要记忆的科目中”

案例研究,了解所有内容的背景

我们的方案提供了一种革命性的技能和知识发展方法。我们的目标是在一个不断变化,竞争激烈和高要求的环境中加强能力建设。

> 和TECH,你可以体验到一种正在动摇世界各地传统大学基础的学习方式”

你将进入一个以重复为基础的学习系统,在整个教学大纲中采用自然和渐进式教学。

学生将通过合作活动和真实案例，学习如何解决真实商业环境中的复杂情况。

一种创新并不同的学习方法

该技术课程是一个密集的教学计划，从零开始，提出了该领域在国内和国际上最苛刻的挑战和决定。由于这种方法，个人和职业成长得到了促进，向成功迈出了决定性的一步。案例法是构成这一内容的技术基础，确保遵循当前经济，社会和职业现实。

> 我们的课程使你准备好在不确定的环境中面对新的挑战，并取得事业上的成功”

在世界顶级计算机科学学校存在的时间里，案例法一直是最广泛使用的学习系统。1912年开发的案例法是为了让法律学生不仅在理论内容的基础上学习法律，案例法向他们展示真实的复杂情况，让他们就如何解决这些问题作出明智的决定和价值判断。1924年，它被确立为哈佛大学的一种标准教学方法。

在特定情况下，专业人士应该怎么做?这就是我们在案例法中面对的问题，这是一种以行动为导向的学习方法。在整个课程中，学生将面对多个真实的案例。他们必须整合所有的知识，研究，论证和捍卫他们的想法和决定。

Re-learning 方法

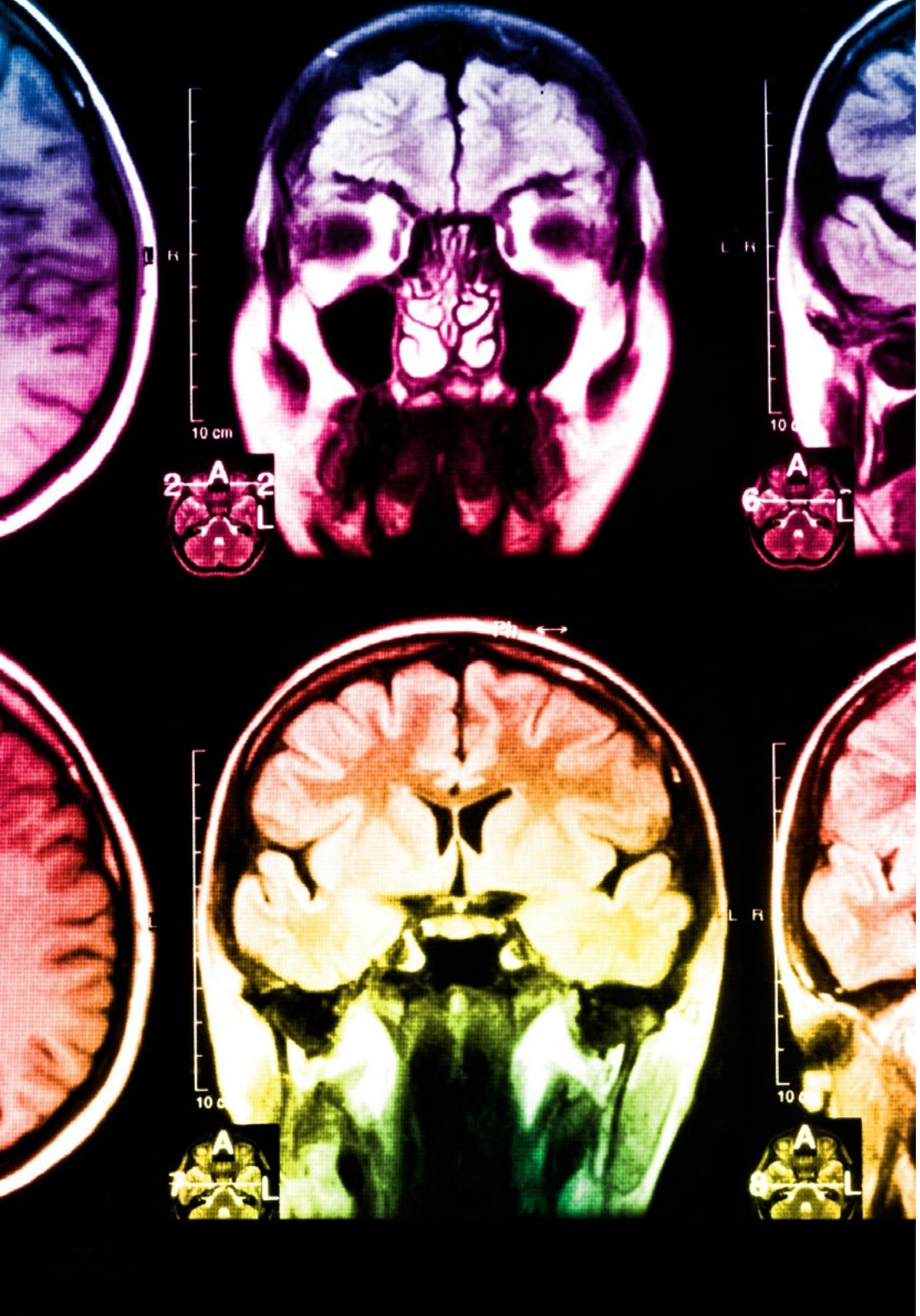
TECH有效地将案例研究方法与基于循环的100%在线学习系统相结合，在每节课中结合了个不同的教学元素。

我们用最好的100%在线教学方法加强案例研究：Re-learning。

> 在2019年，我们取得了世界上所有西班牙语在线大学中最好的学习成绩。

在TECH，你将用一种旨在培训未来管理人员的尖端方法进行学习。这种处于世界教育学前沿的方法被称为 Re-learning。

我校是唯一获准使用这一成功方法的西班牙语大学。2019年，我们成功地提高了学生的整体满意度（教学质量,材料质量,课程结构,目标……），与西班牙语最佳在线大学的指标相匹配。



在我们的方案中，学习不是一个线性的过程，而是以螺旋式的方式发生（学习，解除学习，忘记和重新学习）。因此，我们将这些元素中的每一个都结合起来。这种方法已经培养了超过65万名大学毕业生，在生物化学，遗传学，外科，国际法，管理技能，体育科学，哲学，法律，工程，新闻，历史，金融市场和工具等不同领域取得了前所未有的成功。所有这些都是在一个高要求的环境中进行的，大学学生的社会经济状况很好，平均年龄为43.5岁。

Re-learning 将使你的学习事半功倍，表现更出色，使你更多地参与到训练中，培养批判精神，捍卫论点和对比意见：直接等同于成功。

从神经科学领域的最新科学证据来看，我们不仅知道如何组织信息，想法，图像y记忆，而且知道我们学到东西的地方和背景，这是我们记住它并将其储存在海马体的根本原因，并能将其保留在长期记忆中。

通过这种方式，在所谓的神经认知背景依赖的电子学习中，我们课程的不同元素与学员发展其专业实践的背景相联系。

该方案提供了最好的教育材料，为专业人士做了充分准备：

学习材料

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

然后，这些内容被应用于视听格式，创造了TECH在线工作方法。所有这些，都是用最新的技术，提供最高质量的材料，供学生使用。

大师课程

有科学证据表明第三方专家观察的有用性。

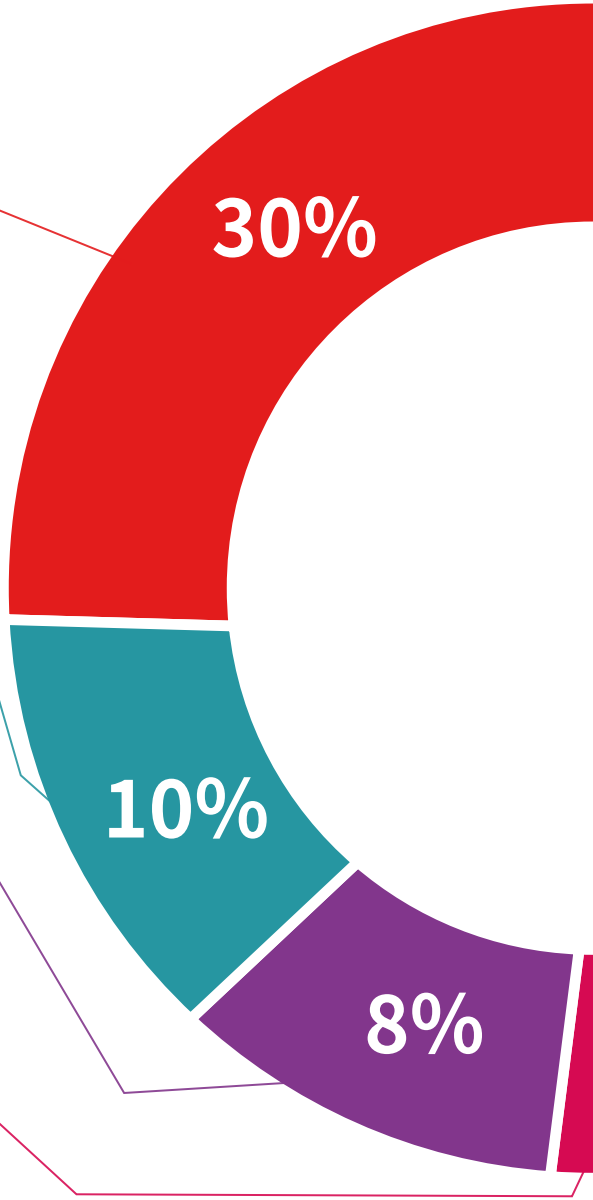
向专家学习可以加强知识和记忆，并为未来的困难决策建立信心。

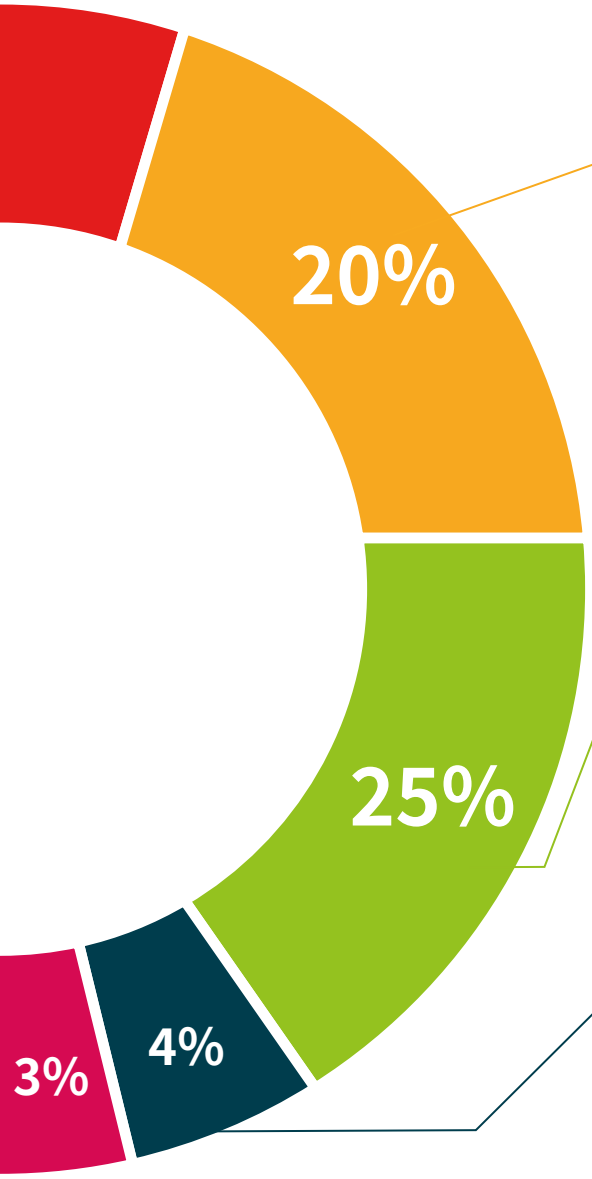
技能和能力的实践

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取得成为专家所需的技能和能力。

延伸阅读

最近的文章，共识文件和国际准则等。在TECH的虚拟图书馆里，学生可以获得他们完成培训所需的一切。





案例研究

他们将完成专门为这个学位选择的最佳案例研究。由国际上最好的专家介绍,分析和辅导案例。

互动式总结

TECH团队以有吸引力和动态的方式将内容呈现在多媒体丸中,其中包括音频,视频,图像,图表和概念图,以强化知识。

这个用于展示多媒体内容的独特教育系统被微软授予“欧洲成功案例”称号。

测试和循环测试

在整个课程中,通过评估和自我评估活动和练习,定期评估和重新评估学习者的知识:通过这种方式,学习者可以看到他/她是如何实现其目标的。

07

学位

Python开发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**Python开发校级硕士**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：**Python开发校级硕士**

模式：**在线**

时长：**7个月**

认证：**ECTS 60**


tech global university

_____ 先生，身份证件号码_____，成功完成并获得了以下学位：

Python开发校级硕士

这是一个持续时间为1,800小时的自主学位，相当于60个ECTS学分，起始日期为yyyy年/mm月/dd日，结束日期为yyyy年/mm月/dd日。

TECH Global University 是一所由安道尔政府于2024年1月31日正式认可的大学，属于欧洲高等教育区（EHEA）。

于2024年2月28日在安道尔首都安道尔城。

Pedro Navarro Illana 博士
校长

Unique TECH Code: AFWORD23S techtitute.com/certificates


Python开发校级硕士

教学大纲的总体分布

科目类型	ECTS 学分
必修(OB)	60
选修 (OP)	0
外部实习 (PR)	0
硕士论文 (TFM)	0
	总计60

教学大纲概况

学期	学科	ECTS	性质
1°	用Python编程	6	必修
1°	使用Python进行高级数据和流控制	6	必修
1°	Python中的面向对象编程(OOP)	6	必修
1°	Python应用程序开发	6	必修
1°	使用Python进行Web和移动开发	6	必修
1°	Python的界面和用户体验	6	必修
1°	使用Python进行数据处理和大数据	6	必修
1°	Python中的数据结构和函数	6	必修
1°	使用NumPy和Pandas在Python中进行数据管理	6	必修
1°	NumPy和Pandas的先进技术和实际应用	6	必修

Pedro Navarro Illana 博士
校长

tech global university

*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。

tech global university

校级硕士
Python开发

- 模式:**在线**
- 时长: **7个月**
- 学位: **TECH Global University**
- 认证: **ECTS 60**
- 课程表:**自由安排时间**
- 考试模式:**在线**

校级硕士
Python开发

tech global university