

Privater Masterstudiengang Entwicklung in Python



Privater Masterstudiengang Entwicklung in Python

- » Modalität: online
- » Dauer: 12 Monate
- » Qualifizierung: TECH Technologische Universität
- » Zeitplan: in Ihrem eigenen Tempo
- » Prüfungen: online

Internetzugang: www.techtitute.com/de/informatik/masterstudiengang/masterstudiengang-entwicklung-python

Index

01

Präsentation

Seite 4

02

Ziele

Seite 8

03

Kompetenzen

Seite 12

04

Kursleitung

Seite 16

05

Struktur und Inhalt

Seite 20

06

Methodik

Seite 30

07

Qualifizierung

Seite 38

01

Präsentation

Python zeichnet sich als vielseitige und effiziente Programmiersprache aus. Erstens erleichtert die klare und lesbare Syntax das Schreiben und Verstehen von Code, was den Entwicklungsprozess beschleunigt und die Fehlerwahrscheinlichkeit verringert. Darüber hinaus bietet die große Vielfalt an verfügbaren Bibliotheken und *Frameworks* den Entwicklern Zugang zu leistungsstarken, bereits vorhandenen Tools, die eine schnellere und effizientere Entwicklung ermöglichen. Der Open-Source-Charakter und die aktive Entwicklergemeinschaft tragen ebenfalls zur ständigen Verbesserung und Aktualisierung der Sprache bei und gewährleisten eine dynamische und moderne Entwicklungsumgebung. Aus diesem Grund hat TECH ein umfassendes Programm entwickelt, das darauf abzielt, Fachleute mit den neuesten Fortschritten in der Python-Entwicklung fortzubilden, und zwar mit einer 100%igen Online-Methodik.



“

Python ist für seine Anpassungsfähigkeit bekannt und wird in Anwendungen von der Webentwicklung bis zur künstlichen Intelligenz eingesetzt. Worauf warten Sie, um sich einzuschreiben?"

Python ist eine Programmiersprache auf hohem Niveau, die unter Informatikern weit verbreitet ist, weil sie über eine breite Palette von Bibliotheken und *Frameworks* verfügt, die gängige Aufgaben vereinfachen und es Entwicklern ermöglichen, sich auf die Logik ihrer Anwendungen zu konzentrieren, anstatt Zeit mit technischen Details zu verschwenden. Ein weiterer großer Vorteil ist ihre Vielseitigkeit, da sie in einer Vielzahl von Umgebungen eingesetzt werden kann, von der Webentwicklung bis hin zur Datenanalyse und zum maschinellen Lernen.

So entstand dieser Private Masterstudiengang in Entwicklung in Python, ein Programm, das den fortgeschrittenen Umgang mit Daten und Typen in Python abdeckt und Bezeichner, Stichwörter, integrale und boolesche Typen sowie die fortgeschrittene Formatierung von *Strings* und Kodierungen erforscht. Darüber hinaus werden Sammlungen wie Tupel, Listen und Wörterbücher sowie Iterationstechniken und Lambda-Funktionen untersucht, um eine solide Grundlage für die Grundlagen der Sprache zu schaffen.

Darüber hinaus werden die Studenten sich mit der Entwicklung von Python-Anwendungen befassen, indem der Schwerpunkt auf *Best Practices* und modernen Methoden liegt. Von der Anwendungsarchitektur bis zur Bereitstellung und Wartung werden Aspekte wie fortgeschrittener Entwurf und Modellierung, Abhängigkeitsmanagement, Entwurfsmuster, *Testing* und *Debugging*, Leistungsoptimierung, Bereitstellung und Verteilungsstrategien behandelt.

Ebenso wird sich der Informatiker mit der Entwicklung von Web- und mobilen Anwendungen mit Python befassen und dabei *Frameworks*, wie Django und Flask sowie die Entwicklung von APIs und Webdiensten behandeln. Darüber hinaus wird der Schwerpunkt auf Schnittstellendesign und Benutzererfahrung (UI/UX) liegen, von der Verwendung von Designtools bis zur Verbesserung der Zugänglichkeit und Benutzerfreundlichkeit. Schließlich werden Datenmanagement und -analyse unter Verwendung von Python und Tools wie NumPy, Pandas und Matplotlib behandelt.

Daher bietet dieser Abschluss eine einzigartige Gelegenheit durch einen vollständig online verfügbaren und anpassbaren Studiengang. Auf diese Weise können Berufstätige ihre Arbeitszeit freier gestalten und ihre täglichen beruflichen und privaten Verpflichtungen besser miteinander in Einklang bringen.

Dieser **Privater Masterstudiengang in Entwicklung in Python** enthält das vollständigste und aktuellste Programm auf dem Markt. Die hervorstechendsten Merkmale sind:

- Die Entwicklung von Fallstudien, die von Experten für Entwicklung in Python vorgestellt werden
- Der anschauliche, schematische und äußerst praxisnahe Inhalt vermittelt theoretische und praktische Informationen zu den Disziplinen, die für die berufliche Praxis unerlässlich sind
- Praktische Übungen, bei denen der Selbstbewertungsprozess zur Verbesserung des Lernens genutzt werden kann
- Sein besonderer Schwerpunkt liegt auf innovativen Methoden
- Theoretische Lektionen, Fragen an den Experten, Diskussionsforen zu kontroversen Themen und individuelle Reflexionsarbeit
- Die Verfügbarkeit des Zugangs zu Inhalten von jedem festen oder tragbaren Gerät mit Internetanschluss



Eine umfassende und spezialisierte Fortbildung in Entwicklung in Python, die Sie auf die realen Herausforderungen in der Welt der Software-Entwicklung vorbereitet"

“

In diesem privaten Masterstudiengang, der zu 100% online angeboten wird, vertiefen Sie Ihre Kenntnisse in den Bereichen Frontend- und Backend-Design, Datenbankmanagement und Veröffentlichungsstrategien in App-Stores"

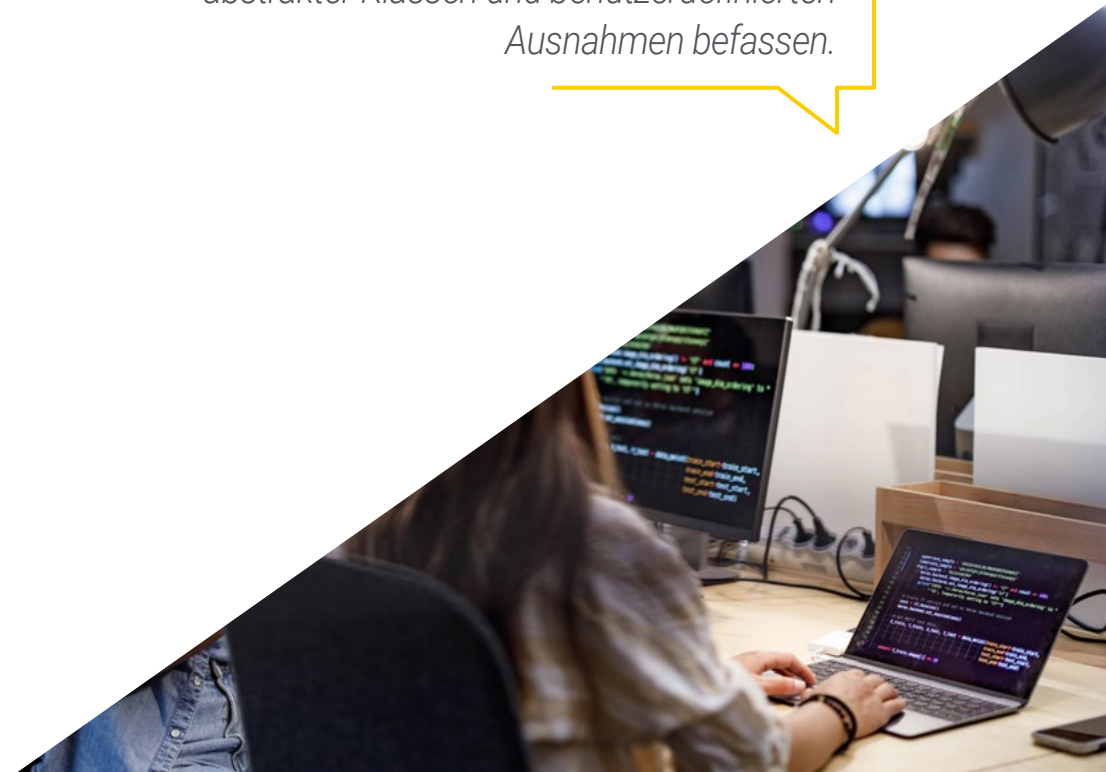
Das Dozententeam des Programms besteht aus Experten des Sektors, die ihre Berufserfahrung in diese Fortbildung einbringen, sowie aus renommierten Fachleuten von führenden Gesellschaften und angesehenen Universitäten.

Die multimedialen Inhalte, die mit der neuesten Bildungstechnologie entwickelt wurden, werden der Fachkraft ein situiertes und kontextbezogenes Lernen ermöglichen, 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 die Fachkraft versuchen muss, die verschiedenen Situationen aus der beruflichen Praxis zu lösen, die während des gesamten Studiengangs gestellt werden. Zu diesem Zweck wird sie von einem innovativen interaktiven Videosystem unterstützt, das von renommierten Experten entwickelt wurde.

Sie werden alles von den Grundlagen wie Variablen und Datentypen bis hin zu fortgeschrittenen Datenvisualisierungstechniken und Strategien zur Leistungs- und Speicheroptimierung behandeln. Schreiben Sie sich jetzt ein!

Sie werden sich mit objektorientierter Programmierung (OOP) sowie mit Themen wie Klassen, Vererbung, Polymorphismus, Erstellung abstrakter Klassen und benutzerdefinierter Ausnahmen befassen.



02 Ziele

Dieser private Masterstudiengang wurde mit dem festen Ziel konzipiert, hochkompetente und spezialisierte Fachleute für das weite Feld der Programmierung in dieser Hochleistungssprache weiterzubilden. Durch umfassende Module erhalten die Studenten einen tiefen Einblick in die Python-Syntax und -Funktionalität, von fortgeschrittener Datenverarbeitung bis hin zur Beherrschung der objektorientierten Programmierung (OOP) und dem effizienten Design von Web- und mobilen Anwendungen. Mit einem fundierten und modernen pädagogischen Ansatz garantiert dieses Programm eine einzigartige Bildungserfahrung, die Informatiker zu gefragten Experten in der Softwareentwicklungsbranche macht. Machen Sie sich bereit, an Ihre Grenzen zu gehen, und begeben Sie sich auf eine Bildungsreise, die nicht nur Ihre Python-Kenntnisse verbessert, sondern Ihnen auch die Werkzeuge an die Hand gibt, um in der dynamischen Welt der Technologieentwicklung zu bestehen und zu führen.



“

Nehmen Sie eine führende Position im Bereich der Softwareentwicklung ein, indem Sie sich mit fortgeschrittenem Wissen und spezialisierten Fähigkeiten ausstatten, um die heutigen technologischen Herausforderungen zu bewältigen"



Allgemeine Ziele

- Vermitteln eines umfassenden Verständnisses von Python
- Fortbilden in der fortgeschrittenen Daten- und Typverarbeitung in Python
- Anwenden der Grundsätze der objektorientierten Programmierung (OOP) in Python
- Fördern der Anwendung bewährter Verfahren und moderner Methoden bei der Softwareentwicklung
- Umfassendes Fortbilden in Web- und mobiler Entwicklung mit Python
- Integrieren von UI/UX-Prinzipien in die Softwareentwicklung
- Fortbilden in der Konfiguration und Nutzung von Werkzeugen und Entwicklungsumgebungen für Daten
- Vertiefen der Verwendung von Datenstrukturen und Funktionen in Python
- Fortbilden in fortgeschrittenen Datenvisualisierungstechniken mit Matplotlib
- Fortbilden in Strategien zur Leistungsoptimierung und zum Data Warehousing



Dank der umfangreichen Bibliothek dieses Studiengangs, die mit den innovativsten Multimedia-Ressourcen gefüllt ist, erhalten Sie das Rüstzeug, um sich in der dynamischen Welt der technologischen Entwicklung hervorzuheben und eine Führungsrolle zu übernehmen"





Spezifische Ziele

Modul 1. Programmierung in Python

- ♦ Einrichten und effektives Nutzen der Python-Entwicklungsumgebung
- ♦ Verstehen von fortgeschrittenen Programmierkonzepten

Modul 2. Fortgeschrittene Daten und Flusskontrolle mit Python

- ♦ Beherrschen der Konventionen und Praktiken für den Umgang mit Bezeichnern und Schlüsselwörtern
- ♦ Anwenden komplexer Datenstrukturen und ihrer Operationen

Modul 3. Objektorientierte Programmierung (OOP) in Python

- ♦ Beherrschen der Erstellung und Verwendung von Klassen und Objekten in Python
- ♦ Anwenden von Vererbung und Polymorphismus in Python

Modul 4. Anwendungsentwicklung in Python

- ♦ Spezialisieren auf fortgeschrittenes Anwendungsdesign und -modellierung
- ♦ Erwerben von Kenntnissen in den Bereichen Optimierung, Einsatz und Wartung von Anwendungen

Modul 5. Web- und mobile Entwicklung mit Python

- ♦ Verwenden beliebter Python-Web- *Frameworks*
- ♦ Vorbereiten für die Entwicklung und Veröffentlichung von mobilen Anwendungen

Modul 6. Benutzeroberfläche und Benutzererfahrung mit Python

- ♦ Unterweisen in responsiven und adaptiven Designtechniken
- ♦ Vorbereiten für die Durchführung von Usability-Tests und die Analyse des Benutzerverhaltens

Modul 7. Datenverarbeitung und *Big Data* mit Python

- ♦ Verwalten von Flusskontrolltechniken und Funktionen für die Datenverarbeitung
- ♦ Fördern bewährter Praktiken in der Python-Programmierung und Fehlerbehandlung

Modul 8. Datenstrukturen und Funktionen in Python

- ♦ Erstellen und Verwenden erweiterter Funktionen
- ♦ Lesen und Schreiben von Dateien und deren Verarbeitung

Modul 9. Datenverwaltung in Python mit NumPy und Pandas

- ♦ Erstellen und Bearbeiten von Arrays mit NumPy
- ♦ Fördern der Kompetenz in der Datenvisualisierung mit Matplotlib

Modul 10. Fortgeschrittene Techniken und praktische Anwendungen in NumPy und Pandas

- ♦ Entwickeln von Fachwissen über das Hochladen und Speichern von Daten in und aus verschiedenen Quellen
- ♦ Fortbilden in fortgeschrittenen Strategien zur Datenbereinigung und -umwandlung

03

Kompetenzen

Das Programm vermittelt nicht nur technisches Wissen, sondern konzentriert sich auch auf die Förderung von Kompetenzen, die die Studenten zu herausragenden Fachleuten im Bereich der Softwareentwicklung machen werden. Dies umfasst die Fähigkeit zur Entwicklung innovativer und effizienter Lösungen durch die Beherrschung von Python, von der fortgeschrittenen Datenverarbeitung bis hin zur Erstellung leistungsstarker Web- und mobilen Anwendungen. Darüber hinaus werden die Informatiker mit speziellen Kenntnissen in objektorientierter Programmierung (OOP), Design von Benutzeroberflächen und Benutzererfahrung (UI/UX) sowie fortgeschrittener Datenanalyse mit Bibliotheken wie NumPy und Pandas ausgestattet.

“

*Die Entwicklung wichtiger Soft Skills wie
kritisches Denken, Problemlösung, Teamarbeit
und Anpassungsfähigkeit wird gefördert“*



Allgemeine Kompetenzen

- Entwickeln praktischer Programmierkenntnisse
- Ausstatten mit fortgeschrittenen Fertigkeiten in der Steuerung des Programmflusses
- Verwalten des Entwurfs und der Implementierung von objektorientierter Software
- Fortbilden in der integralen Entwicklung von Anwendungen in Python
- Beherrschen der Gestaltung und Verwaltung von Web- und mobilen Anwendungen
- Steuern des Designs der Benutzeroberfläche und der Benutzererfahrung mit Python
- Entwickeln von Kompetenzen in der Datenverarbeitung und -analyse mit Python
- Erwerben von Fähigkeiten in fortgeschrittener Dateiverarbeitung und Modellierung in Python
- Entwickeln von Fähigkeiten in fortgeschrittener Datenverarbeitung mit NumPy und Pandas
- Vertiefen der fortgeschrittenen Datenverarbeitung mit NumPy und Pandas

“

Sie werden sich realen Herausforderungen stellen und sich auf einem wettbewerbsorientierten und sich ständig weiterentwickelnden Arbeitsmarkt behaupten. Setzen Sie auf TECH!"





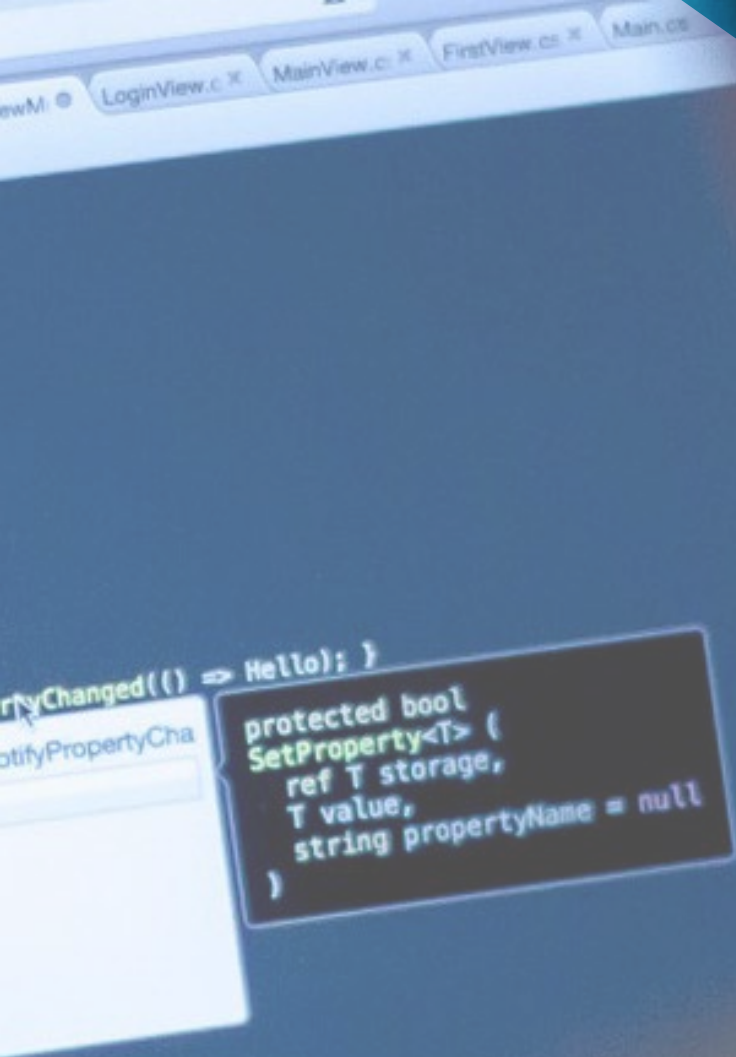
Spezifische Kompetenzen

- ♦ Fortbilden in fortgeschrittener Datenverarbeitung in Python
- ♦ Beherrschen des fortgeschrittenen Einsatzes von Funktionen in Python
- ♦ Implementieren fortgeschrittener Konzepte der objektorientierten Programmierung (OOP) wie abstrakte Klassen und benutzerdefinierte Ausnahmen
- ♦ Kontrollieren von Tests und *Debugging*
- ♦ Entwickeln von APIs und Webdiensten
- ♦ Beherrschen der Verwendung von UI/UX-Design-Tools mit Python
- ♦ Verwenden wichtiger Bibliotheken für Daten in Python
- ♦ Praktisches Anwenden verschiedener Datenstrukturen
- ♦ Verwenden von Pandas für die Verwaltung strukturierter Daten
- ♦ Fördern von Kompetenzen bei der Analyse und Bearbeitung von Zeitreihen und komplexen Daten

04 Kursleitung

Die Dozenten dieses privaten Masterstudiengangs sind Branchenexperten mit umfassender praktischer Erfahrung in der Softwareentwicklung und einer nachweislichen Erfolgsbilanz von Projekten. Durch die Verbindung von Theorie und praktischer Anwendung vermitteln diese Mentoren nicht nur fortgeschrittene Python-Kenntnisse, sondern geben auch ihre Erfahrungen aus der Praxis weiter, so dass die Studenten einen wertvollen Einblick in aktuelle *Best Practices* und Trends in der Branche erhalten. Darüber hinaus zeichnet sich der pädagogische Ansatz des Studiengangs durch sein Engagement aus, den Studenten die modernste technische Weiterbildung zukommen zu lassen, damit sie sich im dynamischen Bereich der Softwareentwicklung behaupten können.





“

Sie werden mit den Besten Erfolg haben und das Wissen und die Fähigkeiten erwerben, die Sie für den Einstieg in die Branche der Python-Entwicklung benötigen"

Leitung



Hr. Matos Rodríguez, Dionis

- ♦ *Data Engineer* bei Wide Agency Sodexo
- ♦ *Data Consultant* bei Tokiota
- ♦ *Data Engineer* bei Devoteam
- ♦ *BI Developer* bei Ibermática
- ♦ *Applications Engineer* bei Johnson Controls
- ♦ *Database Developer* bei Suncapital España
- ♦ *Senior Web Developer* bei Deadlock Solutions
- ♦ *QA Analyst* bei Metaconcept
- ♦ Masterstudiengang in *Big Data & Analytics* von der EAE Business School
- ♦ Masterstudiengang in Systemanalyse und -design
- ♦ Hochschulabschluss in Computertechnik an der Universität APEC

Professoren

Hr. Villar Valor, Javier

- ◆ Direktor und Gründungspartner von Impulsa2
- ◆ *Chief Operations Officer (COO)* bei Summa Insurance Brokers
- ◆ Direktor für Transformation und betriebliche Exzellenz bei Johnson Controls
- ◆ Masterstudiengang in Professionelles *Coaching*
- ◆ Executive MBA der Emlyon Business School, Frankreich
- ◆ Masterstudiengang in Qualitätsmanagement von EOI
- ◆ Hochschulabschluss in Computertechnik an der Universität Acción Pro-Educación y Cultura (UNAPEC)

Hr. Gil Contreras, Armando

- ◆ *Lead Big Data Scientist* bei Johnson Controls
- ◆ *Data Scientist-Big Data* bei Opensistemas S.A.
- ◆ Wirtschaftsprüfer bei Creatividad y Tecnología S.A. (CYTSA)
- ◆ Wirtschaftsprüfer für den öffentlichen Sektor bei PricewaterhouseCoopers Auditors
- ◆ Masterstudiengang in *Data Science* am Universitätszentrum für Technologie und Kunst
- ◆ Masterstudiengang MBA in Internationale Beziehungen und Wirtschaft am Finanzstudienzentrum (CEF)
- ◆ Hochschulabschluss in Wirtschaftswissenschaften am Technologischen Institut von Santo Domingo

Fr. Gil Contreras, Milagros

- ◆ *Content Creator* bei MPCTech LLC
- ◆ Projektleiterin
- ◆ *Freelance IT Writer*
- ◆ MBA an der Universität Complutense von Madrid
- ◆ Hochschulabschluss in Betriebswirtschaftslehre am Technischen Institut von Santo Domingo

Fr. Delgado Feliz, Benedit

- ◆ Verwaltungsassistentin und Operatorin für elektronische Überwachung bei der Nationalen Drogenkontrollbehörde (DNCD)
- ◆ Kundendienst bei Cáceres y Equipos
- ◆ Reklamationen und Kundendienst bei Express Parcel Services (EPS)
- ◆ Spezialistin für Microsoft Office von der Nationalen Schule für Informatik
- ◆ Soziale Kommunikatorin von der Katholischen Universität Santo Domingo



Nutzen Sie die Gelegenheit, sich über die neuesten Fortschritte auf diesem Gebiet zu informieren und diese in Ihrer täglichen Praxis anzuwenden“

05 Struktur und Inhalt

Der Lehrplan wurde sorgfältig erstellt, um den Studenten eine umfassende Erfahrung zu vermitteln. Von einem tiefen Eintauchen in die Syntax und die fortgeschrittenen Funktionen von Python bis hin zur Spezialisierung in objektorientierter Programmierung (OOP), dem Design von Web- und mobilen Anwendungen und der fachkundigen Datenverarbeitung mit Bibliotheken wie NumPy und Pandas wird jedes Modul sorgfältig strukturiert sein, um solide Kenntnisse und praktische Fähigkeiten zu vermitteln. Darüber hinaus werden wichtige Themen wie das Design der Benutzeroberfläche und die Benutzererfahrung (UI/UX), fortgeschrittene Datenverarbeitung sowie Leistungs- und Speicheroptimierung erforscht.



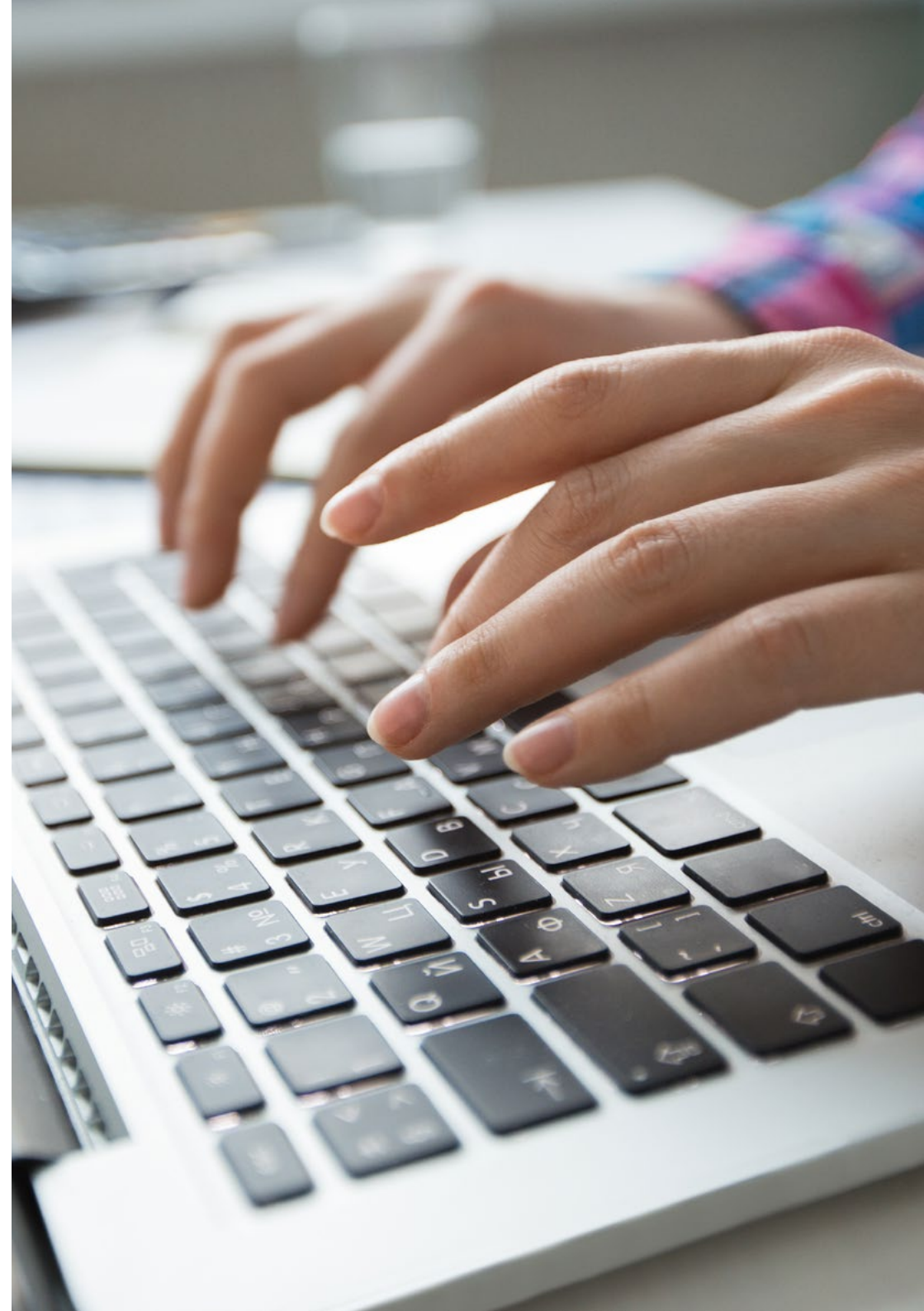


“

Sie werden nicht nur Python vollständig beherrschen, sondern auch darauf vorbereitet sein, dynamische, branchenbezogene technologische Herausforderungen selbstbewusst anzugehen"

Modul 1. Programmierung in Python

- 1.1. Erstellen und Ausführen von Python-Programmen
 - 1.1.1. Konfiguration der Entwicklungsumgebung
 - 1.1.2. Ausführung von *Scripts* in Python
 - 1.1.3. Integrierte Entwicklungswerkzeuge (IDEs)
- 1.2. Daten in Python
 - 1.2.1. Primitive Typen (*int*, *float*, *str*)
 - 1.2.2. *Casting* und Konvertierung von Datentypen in Python
 - 1.2.3. Unveränderlichkeit und Datenspeicherung in Python
- 1.3. Referenzen auf Objekte in Python
 - 1.3.1. Referenzen im Speicher
 - 1.3.2. Identität vs. Gleichheit
 - 1.3.3. Referenzenmanagement und Abfallsammlung
- 1.4. Datenerhebung in Python
 - 1.4.1. Allgemeine Listen und Operationen
 - 1.4.2. Tupel und ihre Unveränderlichkeit
 - 1.4.3. Wörterbücher und Datenzugriff
- 1.5. Logische Operationen in Python
 - 1.5.1. Boolesche Operatoren
 - 1.5.2. Bedingte Ausdrücke
 - 1.5.3. *Short-Circuit Evaluation*
- 1.6. Arithmetische Operatoren in Python
 - 1.6.1. Arithmetische Operationen in Python
 - 1.6.2. Divisionsoperatoren
 - 1.6.3. Vorrangigkeit und Assoziativität
- 1.7. Python-Eingang/Ausgang
 - 1.7.1. Lesen von Daten aus der Standardeingabe
 - 1.7.2. Schreiben von Daten auf die Standardausgabe
 - 1.7.3. Dateiverwaltung
- 1.8. Erstellen und Aufrufen von Funktionen in Python
 - 1.8.1. Syntax der Funktionen
 - 1.8.2. Parameter und Argumente
 - 1.8.3. Rückgabewerte und anonyme Funktionen



- 1.9. Verwendung von *Strings* in Python
 - 1.9.1. Bearbeitung und Formatierung von *Strings*
 - 1.9.2. Gemeinsame *String*-Methoden
 - 1.9.3. Interpolation und *F-Strings*
- 1.10. Fehler- und Ausnahmebehandlung in Python
 - 1.10.1. Häufige Arten von Ausnahmen
 - 1.10.2. *Try-Except*-Blöcke
 - 1.10.3. Erstellen von benutzerdefinierten Ausnahmen

Modul 2. Fortgeschrittene Daten und Flusskontrolle mit Python

- 2.1. Bezeichner und Schlüsselwörter in Python
 - 2.1.1. Regeln für Variablennamen
 - 2.1.2. Reservierte Wörter in Python
 - 2.1.3. Konventionen der Nomenklatur
- 2.2. Ganzzahlige und boolesche Typen in Python
 - 2.2.1. Ganzzahlen (Integers)
 - 2.2.2. Boolean-spezifische Operationen
 - 2.2.3. Umrechnungen und Darstellungen
- 2.3. Fließkommazahlen und komplexe Zahlen in Python
 - 2.3.1. Genauigkeit und Darstellung
 - 2.3.2. Fließkommaoperationen
 - 2.3.3. Verwendung von komplexen Zahlen in Berechnungen
- 2.4. *String*-Formatierung und Kodierungen in Python
 - 2.4.1. Erweiterte Formatierungsmethoden
 - 2.4.2. *Unicode* und UTF-8 Kodierungen
 - 2.4.3. Arbeiten mit Sonderzeichen
- 2.5. Sammlungen: Tupel, Listen und Wörterbücher in Python
 - 2.5.1. Vergleich und Kontrast zwischen den Typen
 - 2.5.2. Typspezifische Methoden
 - 2.5.3. Effizienz und Auswahl des geeigneten Typs
- 2.6. *Sets* und *Frozen Sets* in Python
 - 2.6.1. Erstellung und Operationen in *Sets*
 - 2.6.2. *Frozen Sets*
 - 2.6.3. Praktische Anwendungen und Leistung

- 2.7. Iterieren und Kopieren von Sammlungen in Python
 - 2.7.1. *for*-Schleifen und Listen-Abstraktion
 - 2.7.2. Oberflächliches vs. Tiefes Kopieren
 - 2.7.3. Iteratoren und Generatoren
- 2.8. Nutzung von *Lambda*-Funktionen in Python
 - 2.8.1. Syntax und Erstellung von *Lambda*-Funktionen
 - 2.8.2. Anwendungen in Filtern und Karten
 - 2.8.3. Beschränkungen und bewährte Verfahren
- 2.9. Kontrollstrukturen: Konditionale Bedingungen und Schleifen in Python
 - 2.9.1. *if-else*- und *elif*-Strukturen
 - 2.9.2. *while*- und *for*-Schleifen
 - 2.9.3. Ablaufsteuerung mit *break*, *continue* und *else*
- 2.10. Fortgeschrittene Python-Funktionen und -Methoden
 - 2.10.1. Rekursive Funktionen
 - 2.10.2. Funktionen höherer Ordnung
 - 2.10.3. Funktionsdekoratoren

Modul 3. Objektorientierte Programmierung (OOP) in Python

- 3.1. Objektorientierte Programmierung (OOP) in Python
 - 3.1.1. Klassen und Objekte
 - 3.1.2. Funktionsdekoratoren
 - 3.1.3. Objektorientierte Programmierung (OOP) in Python
- 3.2. Erstellen von Klassen und Objekten in Python
 - 3.2.1. OOP-Klassen in Python
 - 3.2.2. Instanziierung und Initialisierungsmethoden
 - 3.2.3. Attribute und Methoden
- 3.3. Attribute und Methoden in Python
 - 3.3.1. Instanz-Attribute vs. Klasse
 - 3.3.2. Instanz-, Klassen- und statische Methoden
 - 3.3.3. Verkapselung und Verheimlichung von Informationen
- 3.4. Vererbung und Polymorphismus in Python
 - 3.4.1. Einfach- und Mehrfachvererbung
 - 3.4.2. Überschreiben und Methodenerweiterung
 - 3.4.3. Polymorphismus und *Duck Typing*

- 3.5. Eigenschaften und Zugriff auf Attribute in Python
 - 3.5.1. *Getters* und *Setters*
 - 3.5.2. Dekorateur *@property*
 - 3.5.3. Zugangskontrolle und Validierung
 - 3.6. Benutzerdefinierte Klassen und Sammlungen in Python
 - 3.6.1. Erstellung von Sammlungsarten
 - 3.6.2. Spezielle Methoden (*__len__*, *__getitem__*)
 - 3.6.3. Kundenspezifische Iteratoren
 - 3.7. Aggregation und Komposition von Klassen in Python
 - 3.7.1. Beziehungen zwischen Klassen
 - 3.7.2. Aggregation vs. Komposition
 - 3.7.3. Verwaltung des Lebenszyklus von Objekten
 - 3.8. Verwendung von Dekoratoren in Python-Klassen
 - 3.8.1. Dekoratoren für Methoden
 - 3.8.2. Klassen-Dekoratoren
 - 3.8.3. Anwendungen und Anwendungsfälle
 - 3.9. Abstrakte Klassen und Methoden in Python
 - 3.9.1. Abstrakte Klassen
 - 3.9.2. Abstrakte Methoden und Implementierung
 - 3.9.3. Verwendung von ABC (*Abstract Base Class*)
 - 3.10. Ausnahmen und Fehlerbehandlung in OOP in Python
 - 3.10.1. Benutzerdefinierte Ausnahmen in Klassen
 - 3.10.2. Behandlung von Ausnahmen in Methoden
 - 3.10.3. Bewährte Praktiken bei Ausnahmen und OOP
 - 4.3. Verwaltung von Abhängigkeiten und Bibliotheken in Python
 - 4.3.1. Paketverarbeitung mit Pip
 - 4.3.2. Nutzung von virtuellen Umgebungen
 - 4.3.3. Auflösung von Konflikten bei Abhängigkeiten
 - 4.4. Entwurfsmuster in der Python-Entwicklung
 - 4.4.1. Kreative, strukturelle und verhaltensbezogene Muster
 - 4.4.2. Praktische Anwendung von Mustern
 - 4.4.3. *Refactoring* und Muster
 - 4.5. Testen und *Debugging* in Python-Anwendungen
 - 4.5.1. *Testing*-Strategien (einheitlich, Integration)
 - 4.5.2. Verwendung von *Test-Frameworks*
 - 4.5.3. *Debugging*-Techniken und Tools
 - 4.6. Sicherheit und Authentifizierung in Python
 - 4.6.1. Sicherheit in Anwendungen
 - 4.6.2. Implementierung von Authentifizierung und Autorisierung
 - 4.6.3. Verhinderung von Schwachstellen
 - 4.7. Optimierung und Leistung von Python-Anwendungen
 - 4.7.1. Leistungsanalyse
 - 4.7.2. Techniken zur Code-Optimierung
 - 4.7.3. Effizientes Ressourcen- und Datenmanagement
 - 4.8. Bereitstellung und Verteilung von Python-Anwendungen
 - 4.8.1. Strategien für die Bereitstellung
 - 4.8.2. Einsatz von Containern und Orchestrieren
 - 4.8.3. Verteilung und kontinuierliche Aktualisierung
 - 4.9. Instandhaltung und Upgrades in Python
 - 4.9.1. Verwaltung des Software-Lebenszyklus
 - 4.9.2. Instandhaltungs- und Refactoring-Strategien
 - 4.9.3. System-Upgrades und -Migration
 - 4.10. Dokumentation und Support in Python
 - 4.10.1. Erstellen einer effektiven Dokumentation
 - 4.10.2. Werkzeuge für die Dokumentation
 - 4.10.3. Strategien für den Support und die Kommunikation mit den Benutzern
- Modul 4. Anwendungsentwicklung in Python**
- 4.1. Python-Anwendungsarchitektur
 - 4.1.1. Software-Entwicklung
 - 4.1.2. Gemeinsame Architekturmuster
 - 4.1.3. Bewertung der Bedürfnisse und Anforderungen
 - 4.2. Entwurf und Modellierung von Anwendungen in Python
 - 4.2.1. Verwendung von UML und Diagrammen
 - 4.2.2. Datenmodellierung und Informationsfluss
 - 4.2.3. SOLID-Grundsätze und modularer Aufbau

Modul 5. Web- und mobile Entwicklung mit Python

- 5.1. Web-Entwicklung mit Python
 - 5.1.1. Struktur und Bestandteile einer Website
 - 5.1.2. Technologien in der Web-Entwicklung
 - 5.1.3. Trends in der Web-Entwicklung
- 5.2. Beliebte Web-Frameworks mit Python
 - 5.2.1. *Django*, *Flask* und andere Optionen
 - 5.2.2. Vergleich und Auswahl von *Frameworks*
 - 5.2.3. Integration mit *Frontend*
- 5.3. *Frontend*-Entwicklung: HTML, CSS und JavaScript mit Python
 - 5.3.1. HTML und CSS
 - 5.3.2. JavaScript und DOM-Manipulation
 - 5.3.3. *Frameworks* und *Frontend*-Bibliotheken
- 5.4. *Backend* und Datenbanken mit Python
 - 5.4.1. *Backend*-Entwicklung mit Python
 - 5.4.2. Relationale und nichtrelationale Datenbankverwaltung
 - 5.4.3. *Backend-Frontend*-Integration
- 5.5. APIs und Webdienste mit Python
 - 5.5.1. Entwurf von *RESTful*-APIs
 - 5.5.2. Umsetzung und Dokumentation von APIs
 - 5.5.3. Verbrauch und Sicherheit in APIs
- 5.6. Mobile Entwicklung mit Python
 - 5.6.1. Mobile Entwicklungsplattformen (nativ, hybrid)
 - 5.6.2. Entwicklungswerkzeuge und -umgebungen
 - 5.6.3. Anpassung von Anwendungen für mobile Geräte
- 5.7. Mobile Entwicklungsplattformen mit Python
 - 5.7.1. Android und IOS
 - 5.7.2. *Frameworks* für die übergreifende Entwicklung
 - 5.7.3. Tests und *Deployment* auf mobilen Geräten
- 5.8. Design und UX in mobilen Anwendungen mit Python
 - 5.8.1. Gestaltung der mobilen Schnittstelle
 - 5.8.2. Benutzerfreundlichkeit und Benutzererfahrung mit Python
 - 5.8.3. Prototyping- und Entwurfswerkzeuge

- 5.9. Mobile Tests und Fehlersuche mit Python
 - 5.9.1. Strategien für das *Testing* in mobilen Geräte
 - 5.9.2. Werkzeuge zur Fehlersuche und Überwachung
 - 5.9.3. Testautomatisierung
- 5.10. Veröffentlichung in Anwendungssshops mit Python
 - 5.10.1. Veröffentlichungsprozess im App Store und bei Google Play
 - 5.10.2. Konformitäts- und Anwendungsrichtlinien
 - 5.10.3. Marketing- und Werbestrategien

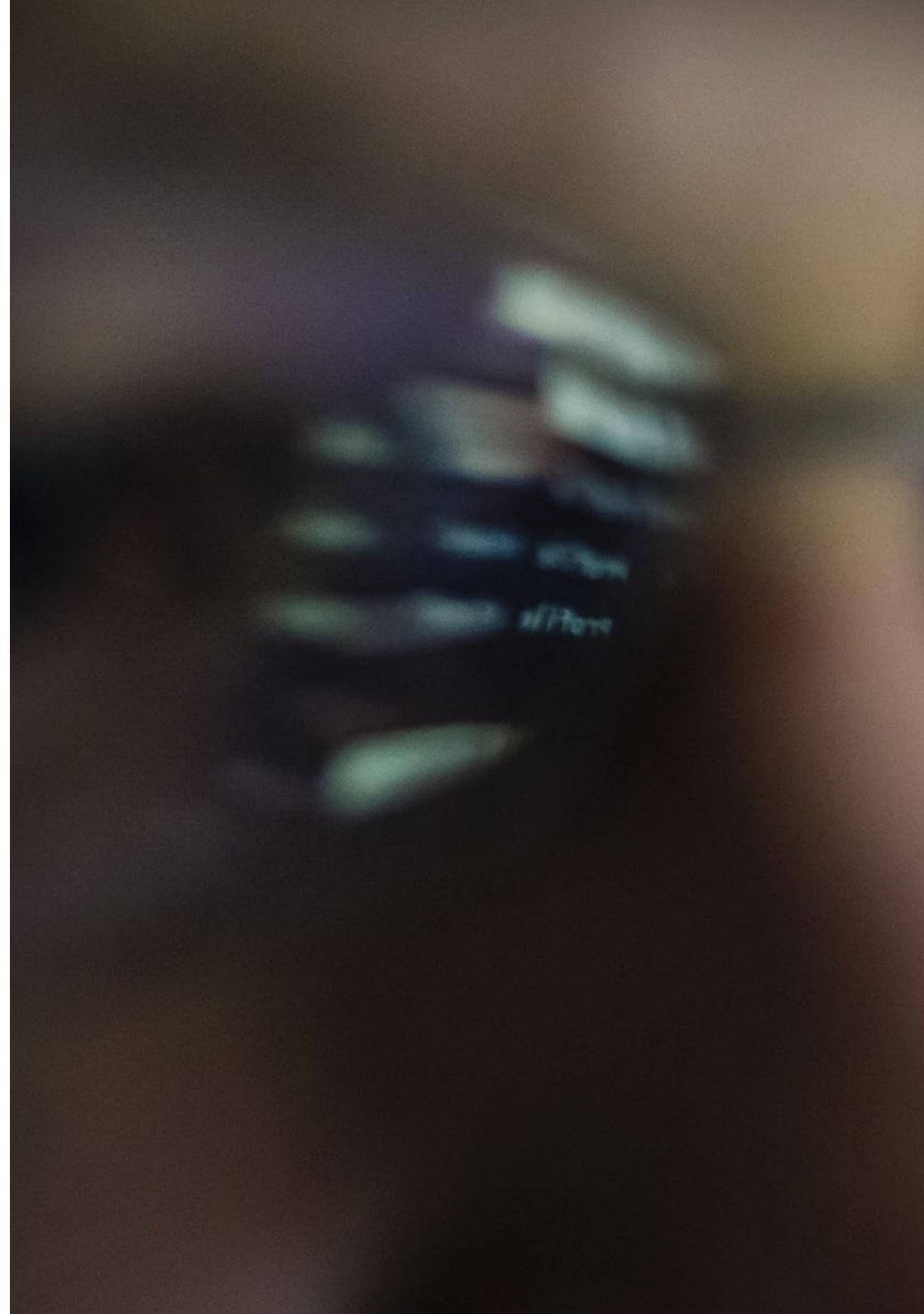
Modul 6. Benutzeroberfläche und Benutzererfahrung mit Python


- 6.1. Design von Benutzeroberflächen mit Python
 - 6.1.1. UI-Design mit Python
 - 6.1.2. Benutzer-Computer-Interaktion mit Python
 - 6.1.3. Benutzerzentriertes Design mit Python
- 6.2. UI/UX-Designwerkzeuge mit Python
 - 6.2.1. Software für Design und Prototyping
 - 6.2.2. Kollaborative Werkzeuge und *Feedback*
 - 6.2.3. Integration des Designs in den Entwicklungsprozess
- 6.3. *Responsive* und *Adaptive Design* mit Python
 - 6.3.1. *Responsive-Design*-Techniken
 - 6.3.2. Anpassung an verschiedene Geräte und Bildschirme
 - 6.3.3. *Testing* und Qualitätssicherung
- 6.4. Animationen und Übergänge mit Python
 - 6.4.1. Erstellen effektiver Animationen mit Python
 - 6.4.2. Werkzeuge und Bibliotheken für die Animation
 - 6.4.3. Auswirkungen auf UX und Leistung
- 6.5. Barrierefreiheit und Benutzerfreundlichkeit mit Python
 - 6.5.1. Barrierefreiheit im Internet
 - 6.5.2. Bewertungsinstrumente und -techniken
 - 6.5.3. Anwendung bewährter Praktiken
- 6.6. *Prototyping* und *Wireframes* mit Python
 - 6.3.1. Erstellung von *Wireframes* und *Mockups*
 - 6.3.2. Werkzeuge für das *Rapid Prototyping*
 - 6.3.3. Tests zur Benutzerfreundlichkeit und *Feedback*

- 6.7. Tests zur Benutzerfreundlichkeit mit Python
 - 6.7.1. Methoden und Techniken der Prüfung der Benutzerfreundlichkeit
 - 6.7.2. Ergebnisorientierte Analyse und Verbesserung
 - 6.7.3. Tools zur Prüfung der Benutzerfreundlichkeit
- 6.8. Analyse des Benutzerverhaltens mit Python
 - 6.8.1. Analyseverfahren und *Tracking*
 - 6.8.2. Datenauswertung und Metriken
 - 6.8.3. Datengestützte kontinuierliche Verbesserung
- 6.9. *Feedback*-basierte Erweiterungen mit Python
 - 6.9.1. Verwaltung und Analyse von *Feedback*
 - 6.9.2. *Feedback*-Zyklen und Kontinuierliche Verbesserung
 - 6.9.3. Strategien zur Umsetzung wirksamer Veränderungen
- 6.10. Zukünftige Trends in UI/UX mit Python
 - 6.10.1. Innovationen und neue Trends
 - 6.10.2. Auswirkungen der neuen Technologien auf UI/UX
 - 6.10.3. Vorbereitungen für die Zukunft des Designs

Modul 7. Datenverarbeitung und *Big Data* mit Python

- 7.1. Verwendung von Python für Daten
 - 7.1.1. Python in der Datenwissenschaft und Analytik
 - 7.1.2. Wichtige Datenbibliotheken
 - 7.1.3. Anwendungen und Beispiele
- 7.2. Konfiguration der Python-Entwicklungsumgebung
 - 7.2.1. Python-Installation und Werkzeuge
 - 7.2.2. Konfiguration von virtuellen Umgebungen
 - 7.2.3. Integrierte Entwicklungswerkzeuge (IDE)
- 7.3. Variablen, Datentypen und Operatoren in Python
 - 7.3.1. Variablen und primitive Datentypen
 - 7.3.2. Datenstrukturen
 - 7.3.3. Arithmetische und logische Operatoren
- 7.4. Flusskontrolle: Konditionalitäten und Schleifen
 - 7.4.1. Konditionale Kontrollstrukturen (*if*, *else*, *elif*)
 - 7.4.2. Schleifen (*for*, *while*) und Flusskontrolle
 - 7.4.3. Listen-Abstraktion und generative Ausdrücke



- 
- 7.5. Funktionen und Modularität mit Python
 - 7.5.1. Verwendung von Funktionen
 - 7.5.2. Parameter, Argumente und Rückgabewerte
 - 7.5.3. Modularität und Wiederverwendung von Code
 - 7.6. Fehler- und Ausnahmebehandlung mit Python
 - 7.6.1. Fehler und Ausnahmen
 - 7.6.2. Ausnahmebehandlung mit *try-except*
 - 7.6.3. Erstellen von benutzerdefinierten Ausnahmen
 - 7.7. *IPython*-Werkzeug
 - 7.7.1. *IPython*-Werkzeug
 - 7.7.2. Verwendung von *IPython* für die Datenanalyse
 - 7.7.3. Unterschiede zum Standard-*Python*-Interpreter
 - 7.8. *Jupyter Notebooks*
 - 7.8.1. *Jupyter Notebooks*
 - 7.8.2. Verwendung von Notebooks für die Datenanalyse
 - 7.8.3. Veröffentlichung von *Jupyter Notebooks*
 - 7.9. Bewährte Praktiken der Python-Programmierung
 - 7.9.1. Stil und Konventionen (PEP 8)
 - 7.9.2. Dokumentation und Anmerkungen
 - 7.9.3. Test- und Fehlerbehebungsstrategien
 - 7.10. Python-Ressourcen und Communities
 - 7.10.1. Online-Ressourcen und Dokumentation
 - 7.10.2. Gemeinschaften und Foren
 - 7.10.3. Lernen und Aktualisieren in Python

Modul 8. Datenstrukturen und Funktionen in Python

- 8.1. Gruppen in Python
 - 8.1.1. Operationen und Methoden
 - 8.1.2. Unterschiede und praktische Anwendung
 - 8.1.3. Iteration und Verständnis
- 8.2. Wörterbücher und ihre Verwendung in Python
 - 8.2.1. Erstellung und Handhabung von Wörterbüchern
 - 8.2.2. Datenzugriff und -verwaltung
 - 8.2.3. Muster und fortgeschrittene Techniken

- 8.3. Listen-Abstraktion und Wörterbücher in Python
 - 8.3.1. Syntax und Beispiele
 - 8.3.2. Effizienz und Lesbarkeit
 - 8.3.3. Praktische Anwendungen
- 8.4. Funktionen für Daten in Python
 - 8.4.1. Erstellung von Funktionen
 - 8.4.2. Geltungsbereich und Namensraum
 - 8.4.3. Anonyme Funktionen und *Lambda*
- 8.5. Funktionsargumente und Rückgabewerte in Python
 - 8.5.1. Positionale und benannte Argumente
 - 8.5.2. Mehrere Rückgabewerte
 - 8.5.3. Variable Argumente und Schlüsselwörter
- 8.6. *Lambda*-Funktionen und Funktionen höherer Ordnung in Python
 - 8.6.1. Verwendung von *Lambda*-Funktionen
 - 8.6.2. *Map*-, *Filter*- und *Reduce*-Funktionen
 - 8.6.3. Datenverarbeitungsanwendungen
- 8.7. Handhabung von Python-Dateien
 - 8.7.1. Lesen und Schreiben von Dateien
 - 8.7.2. Handhabung von Binär- und Textdateien
 - 8.7.3. Bewährte Praktiken und Ausnahmebehandlung
- 8.8. Lesen und Schreiben von Text- und Binärdateien in Python
 - 8.8.1. Dateiformate und Kodierung
 - 8.8.2. Umgang mit großen Dateien
 - 8.8.3. Serialisierung und Deserialisierung (*JSON*, *pickle*)
- 8.9. Kontexte und Dateioperationen
 - 8.9.1. Verwendung des Kontextmanagers (*with*)
 - 8.9.2. Techniken der Dateiverarbeitung
 - 8.9.3. Sicherheit und Fehlerbehandlung
- 8.10. Python-Modellierungsbibliotheken
 - 8.10.1. *Scikit-learn*
 - 8.10.2. *TensorFlow*
 - 8.10.3. *PyTorch*

Modul 9. Datenverwaltung in Python mit NumPy und Pandas

- 9.1. Erstellen und Manipulieren von *Arrays* in NumPy
 - 9.1.1. NumPy
 - 9.1.2. Grundlegende Operationen mit *Arrays*
 - 9.1.3. Manipulation und Transformation von *Arrays*
- 9.2. Vektorisierte Operationen mit *Arrays*
 - 9.2.1. Vektorisierung
 - 9.2.2. Universelle Funktionen (*ufunc*)
 - 9.2.3. Effizienz und Leistung
- 9.3. Indizierung und Segmentierung in NumPy
 - 9.3.1. Zugang zu Elementen und *Slicing*
 - 9.3.2. Erweiterte und boolesche Indizierung
 - 9.3.3. Neuordnung und Auswahl
- 9.4. Pandas-Serien und *DataFrames*
 - 9.4.1. Pandas
 - 9.4.2. Datenstrukturen in Pandas
 - 9.4.3. Handhabung von *DataFrames*
- 9.5. Indizierung und Auswahl in Pandas
 - 9.5.1. Zugang zu Daten in Serien und *DataFrames*
 - 9.5.2. Auswahl- und Filtermethoden
 - 9.5.3. Einsatz von *loc* und *iloc*
- 9.6. Operationen mit Pandas
 - 9.6.1. Arithmetische Operationen und Ausrichtung
 - 9.6.2. Aggregations- und Statistikfunktionen
 - 9.6.3. Transformationen und Anwendung von Funktionen
- 9.7. Handhabung unvollständiger Daten in Pandas
 - 9.7.1. Erkennung und Behandlung von Nullwerten
 - 9.7.2. Ausfüllen und Löschen von unvollständigen Daten
 - 9.7.3. Strategien für den Umgang mit unvollständigen Daten
- 9.8. Funktionen und Anwendungen in Pandas
 - 9.8.1. Verkettung und Zusammenführung von Daten
 - 9.8.2. Gruppierung und Aggregation (*groupby*)
 - 9.8.3. *Pivot Tables* und *Crosstabs*

- 9.9. Visualisierung mit Matplotlib
 - 9.9.1. Matplotlib
 - 9.9.2. Erstellung und Anpassung von Diagrammen
 - 9.9.3. Integration mit Pandas
- 9.10. Anpassung von Diagrammen in Matplotlib
 - 9.10.1. Stile und Konfigurationen
 - 9.10.2. Fortgeschrittene Diagramme (*scatter*, *bar* usw.)
 - 9.10.3. Erstellung komplexer Visualisierungen

Modul 10. Fortgeschrittene Techniken und praktische Anwendungen in NumPy und Pandas

- 10.1. Laden von Daten aus verschiedenen Quellen
 - 10.1.1. Import aus CSV, Excel und Datenbanken
 - 10.1.2. Lesen von Daten aus APIs und dem Internet
 - 10.1.3. Strategien zur Verwaltung großer Datenmengen
- 10.2. Datenspeicherung in Python
 - 10.2.1. Export in verschiedene Formate
 - 10.2.2. Effizienz der Speicherung
 - 10.2.3. Datensicherheit und Datenschutz
- 10.3. Datenbereinigungsstrategien in Python
 - 10.3.1. Identifizierung und Korrektur von Inkonsistenzen
 - 10.3.2. Normalisierung und Umwandlung von Daten
 - 10.3.3. Automatisierung von Bereinigungsprozessen
- 10.4. Erweiterte Datentransformation in Pandas
 - 10.4.1. Handhabung und Transformationstechniken
 - 10.4.2. Kombinieren und Umstrukturieren von *DataFrames*
 - 10.4.3. Verwendung regulärer Ausdrücke in Pandas
- 10.5. Kombination von *DataFrames* in Pandas
 - 10.5.1. *Merge*, *Join* und Verkettung
 - 10.5.2. Konflikt- und Schlüsselmanagement
 - 10.5.3. Effiziente Kombinationsstrategien
- 10.6. Erweiterte Transformation und Pivotierung von Daten in Pandas
 - 10.6.1. *Pivot* und *Melt*
 - 10.6.2. *Reshape*-Techniken und Transponieren
 - 10.6.3. Anwendungen in der Datenanalyse
- 10.7. Zeitreihen in Pandas
 - 10.7.1. Zeit- und Datumsverwaltung
 - 10.7.2. *Resampling* und *Window Functions*
 - 10.7.3. Analyse von Trends und Saisonalität
- 10.8. Erweiterte Indexverwaltung in Pandas
 - 10.8.1. Mehrstufige und hierarchische Indizes
 - 10.8.2. Fortgeschrittene Auswahl und Manipulation
 - 10.8.3. Optimierung von Abfragen
- 10.9. Strategien zur Leistungsoptimierung
 - 10.9.1. Verbesserungen bei Geschwindigkeit und Effizienz
 - 10.9.2. Verwendung von Cython und Numba
 - 10.9.3. Parallelisierung und verteilte Verarbeitung
- 10.10. Praktische Projekte zur Datenmanipulation
 - 10.10.1. Entwicklung von Anwendungsbeispielen aus der Praxis
 - 10.10.2. Integration von Python-Techniken
 - 10.10.3. Strategien zur Lösung komplexer Datenprobleme



Dieses innovative Programm passt sich an Sie an, um Sie mit den neuesten Trends und Technologien auf dem Laufenden zu halten und sicherzustellen, dass Sie an der Spitze der Innovation in der Softwareentwicklung stehen"

06

Methodik

Dieses Fortbildungsprogramm bietet eine andere Art des Lernens. Unsere Methodik wird durch eine zyklische Lernmethode entwickelt: **das Relearning**.

Dieses Lehrsystem wird z. B. an den renommiertesten medizinischen Fakultäten der Welt angewandt und wird von wichtigen Publikationen wie dem **New England Journal of Medicine** als eines der effektivsten angesehen.





Entdecken Sie Relearning, ein System, das das herkömmliche lineare Lernen hinter sich lässt und Sie durch zyklische Lehrsysteme führt: eine Art des Lernens, die sich als äußerst effektiv erwiesen hat, insbesondere in Fächern, die Auswendiglernen erfordern"

Fallstudie zur Kontextualisierung aller Inhalte

Unser Programm bietet eine revolutionäre Methode zur Entwicklung von Fähigkeiten und Kenntnissen. Unser Ziel ist es, Kompetenzen in einem sich wandelnden, wettbewerbsorientierten und sehr anspruchsvollen Umfeld zu stärken.

“

Mit TECH werden Sie eine Art des Lernens erleben, die an den Grundlagen der traditionellen Universitäten auf der ganzen Welt rüttelt"



Sie werden Zugang zu einem Lernsystem haben, das auf Wiederholung basiert, mit natürlichem und progressivem Unterricht während des gesamten Lehrplans.



Der Student wird durch gemeinschaftliche Aktivitäten und reale Fälle lernen, wie man komplexe Situationen in realen Geschäftsumgebungen löst.

Eine innovative und andersartige Lernmethode

Dieses TECH-Programm ist ein von Grund auf neu entwickeltes, intensives Lehrprogramm, das die anspruchsvollsten Herausforderungen und Entscheidungen in diesem Bereich sowohl auf nationaler als auch auf internationaler Ebene vorsieht. Dank dieser Methodik wird das persönliche und berufliche Wachstum gefördert und ein entscheidender Schritt in Richtung Erfolg gemacht. Die Fallmethode, die Technik, die diesem Inhalt zugrunde liegt, gewährleistet, dass die aktuellste wirtschaftliche, soziale und berufliche Realität berücksichtigt wird.

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

Die Fallmethode ist das am weitesten verbreitete Lernsystem an den besten Informatikschulen der Welt, seit es sie gibt. Die Fallmethode wurde 1912 entwickelt, damit Jurastudenten das Recht nicht nur auf der Grundlage theoretischer Inhalte erlernen. Sie bestand darin, ihnen reale komplexe Situationen zu präsentieren, damit sie fundierte Entscheidungen treffen und Werturteile darüber fällen konnten, wie diese zu lösen sind. Sie wurde 1924 als Standardlehrmethode in Harvard etabliert.

Was sollte eine Fachkraft in einer bestimmten Situation tun? Mit dieser Frage konfrontieren wir Sie in der Fallmethode, einer handlungsorientierten Lernmethode. Während des gesamten Kurses werden die Studenten mit mehreren realen Fällen konfrontiert. Sie müssen ihr gesamtes Wissen integrieren, recherchieren, argumentieren und ihre Ideen und Entscheidungen verteidigen.

Relearning Methodology

TECH kombiniert die Methodik der Fallstudien effektiv mit einem 100%igen Online-Lernsystem, das auf Wiederholung basiert und in jeder Lektion verschiedene didaktische Elemente kombiniert.

Wir ergänzen die Fallstudie mit der besten 100%igen Online-Lehrmethode: Relearning.

*Im Jahr 2019 erzielten wir die besten
Lernergebnisse aller spanischsprachigen
Online-Universitäten der Welt.*

Bei TECH lernen Sie mit einer hochmodernen Methodik, die darauf ausgerichtet ist, die Führungskräfte der Zukunft zu spezialisieren. Diese Methode, die an der Spitze der weltweiten Pädagogik steht, wird Relearning genannt.

Unsere Universität ist die einzige in der spanischsprachigen Welt, die für die Anwendung dieser erfolgreichen Methode zugelassen ist. Im Jahr 2019 ist es uns gelungen, die Gesamtzufriedenheit unserer Studenten (Qualität der Lehre, Qualität der Materialien, Kursstruktur, Ziele...) in Bezug auf die Indikatoren der besten spanischsprachigen Online-Universität zu verbessern.



In unserem Programm ist das Lernen kein linearer Prozess, sondern erfolgt in einer Spirale (lernen, verlernen, vergessen und neu lernen). Daher wird jedes dieser Elemente konzentrisch kombiniert. Mit dieser Methode wurden mehr als 650.000 Hochschulabsolventen mit beispiellosem Erfolg in so unterschiedlichen Bereichen wie Biochemie, Genetik, Chirurgie, internationales Recht, Managementfähigkeiten, Sportwissenschaft, Philosophie, Recht, Ingenieurwesen, Journalismus, Geschichte, Finanzmärkte und -instrumente fortgebildet. Dies alles in einem sehr anspruchsvollen Umfeld mit einer Studentenschaft mit hohem sozioökonomischem Profil und einem Durchschnittsalter von 43,5 Jahren.

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.

Nach den neuesten wissenschaftlichen Erkenntnissen der Neurowissenschaften wissen wir nicht nur, wie wir Informationen, Ideen, Bilder und Erinnerungen organisieren, sondern auch, dass der Ort und der Kontext, in dem wir etwas gelernt haben, von grundlegender Bedeutung dafür sind, dass wir uns daran erinnern und es im Hippocampus speichern können, um es in unserem Langzeitgedächtnis zu behalten.

Auf diese Weise sind die verschiedenen Elemente unseres Programms im Rahmen des so genannten Neurocognitive Context-Dependent E-Learning mit dem Kontext verbunden, in dem der Teilnehmer seine berufliche Praxis entwickelt.



Dieses Programm bietet die besten Lehrmaterialien, die sorgfältig für Fachleute aufbereitet sind:



Studienmaterial

Alle didaktischen Inhalte werden von den Fachleuten, 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 das audiovisuelle Format angewendet, um die Online-Arbeitsmethode von TECH zu schaffen. All dies mit den neuesten Techniken, die in jedem einzelnen der Materialien, die dem Studenten zur Verfügung gestellt werden, qualitativ hochwertige Elemente bieten.



Meisterklassen

Die Nützlichkeit der Expertenbeobachtung ist wissenschaftlich belegt.

Das sogenannte Learning from an Expert festigt das Wissen und das Gedächtnis und schafft Vertrauen für zukünftige schwierige Entscheidungen.



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.



Weitere Lektüren

Aktuelle Artikel, Konsensdokumente und internationale Leitfäden, u. a. In der virtuellen Bibliothek von TECH hat der Student Zugang zu allem, was er für seine Fortbildung benötigt.





Case Studies

Sie werden eine Auswahl der besten Fallstudien vervollständigen, die speziell für diese Qualifizierung ausgewählt wurden. Die Fälle werden von den besten Spezialisten der internationalen Szene präsentiert, analysiert und betreut.



Interaktive Zusammenfassungen

Das TECH-Team präsentiert die Inhalte auf attraktive und dynamische Weise in multimedialen Pillen, die Audios, Videos, Bilder, Diagramme und konzeptionelle Karten enthalten, um das Wissen zu vertiefen.

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



Testing & Retesting

Die Kenntnisse des Studenten werden während des gesamten Programms regelmäßig durch Bewertungs- und Selbsteinschätzungsaktivitäten und -übungen beurteilt und neu bewertet, so dass der Student überprüfen kann, wie er seine Ziele erreicht.



07

Qualifizierung

Der Privater Masterstudiengang in Entwicklung in Python garantiert neben der präzisesten und aktuellsten Fortbildung auch den Zugang zu einem von der TECH Technologischen Universität ausgestellten Diplom.



“

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

Dieser **Privater Masterstudiengang in Entwicklung in Python** enthält das vollständigste und aktuellste Programm auf dem Markt.

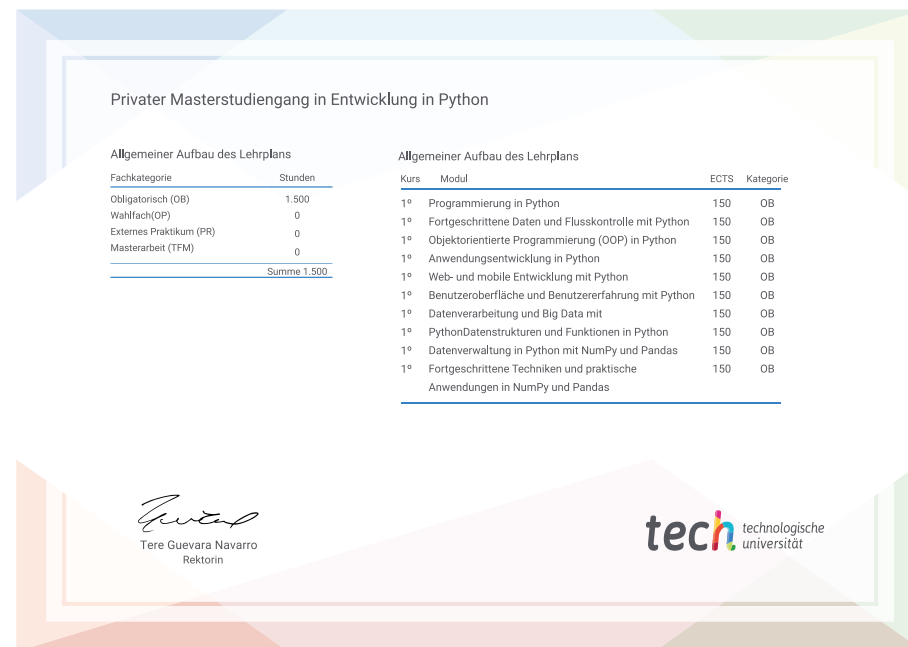
Sobald der Student die Prüfungen bestanden hat, erhält er/sie per Post* mit Empfangsbestätigung das entsprechende Diplom, ausgestellt von der **TECH Technologischen Universität**.

Das von **TECH Technologische Universität** ausgestellte Diplom drückt die erworbene Qualifikation aus und entspricht den Anforderungen, die in der Regel von Stellenbörsen, Auswahlprüfungen und Berufsbildungsausschüssen verlangt werden.

Titel: **Privater Masterstudiengang in Entwicklung in Python**

Modalität: **online**

Dauer: **12 Monate**



*Haager Apostille. Für den Fall, dass der Student die Haager Apostille für sein Papierdiplom beantragt, wird TECH EDUCATION 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 instituten
virtuelles Klassenzimmer sprachen

tech technologische
universität

Privater
Masterstudiengang
Entwicklung in Python

- » Modalität: online
- » Dauer: 12 Monate
- » Qualifizierung: TECH Technologische Universität
- » Zeitplan: in Ihrem eigenen Tempo
- » Prüfungen: online

Privater Masterstudiengang Entwicklung in Python