



Datos Generales

1. Nombre de la Asignatura	2. Nivel de formación	3. Clave de la Asignatura
Programación I	Maestría	IH599
4. Prerrequisitos	5. Área de Formación	6. Departamento
Ninguno	Básica común obligatoria	Sistemas de información
7. Modalidad	8. Tipo de Asignatura:	
Presencial	Curso-Taller	
9. Carga Horaria 60 horas		
Teoría: 40 horas	Práctica: 20 horas	Total: 60
		Créditos: 7
10. Trayectoria de la asignatura		

Contenido del Programa

11. Presentación

El maestrante en Ciencia de datos, debe tener bases sólidas de programación que le permitan desarrollar sus propios sistemas de análisis de datos, a la par de ser capaz de extraer la información de diferentes repositorios (bases de datos y/o archivos), gestionar su código con sistemas de control de versiones (Git), y probar y depurar su propio código.

Con este curso el futuro maestro en Ciencia de datos, obtendrá las habilidades necesarias para dominar el lenguaje de programación Python

12.- Objetivos del programa

Objetivo General

Adquirir las bases del lenguaje de programación Python que le permitan al alumno desarrollar y probar cualquier tipo de programas en Python pudiendo hacerlo de manera colaborativa.

13.-Contenido

Contenido temático

Unidad 1 Arquitectura de Python

Unidad 2 Fundamentos de programación en Python

Unidad 3 Funciones de cadena

Unidad 4 Archivos

Unidad 5 Estructuras de datos en Python

Unidad 6 Expresiones regulares

Unidad 7 Sockets y servicios web

Unidad 8 Bases datos SQL

Unidad 9 Herramientas para depuración y pruebas de código

Unidad 10 Gestión de repositorios de código para control de versiones y desarrollo colaborativo con Git y GitHub

Contenido desarrollado

Unidad 1 Arquitectura de Python

Objetivo particular de la unidad: Conocer como funciona Python a bajo nivel para aprovechar el lenguaje de programación al máximo.

Desarrollo:

1. Arquitectura de Python
 - 1.1. Lenguajes de programación
 - 1.1.1. Lenguajes compilados
 - 1.1.2. Lenguajes interpretados
 - 1.1.3. Lenguajes tipados y lenguajes no tipados
 - 1.2. Historia de Python
 - 1.3. Versiones de Python
 - 1.3.1. 2.x
 - 1.3.2. 3.x
 - 1.4. Chevron vs archivos
 - 1.5. Python Virtual Machine
 - 1.5.1. CPython y Cython
 - 1.5.2. Jython
 - 1.5.3. IronPython
 - 1.5.4. Pypy
 - 1.6. Python en la actualidad

Unidad 2 Fundamentos de programación en Python

Objetivo particular de la unidad: Adquirir las competencias fundamentales de programación en Python

Desarrollo:

2. Fundamentos de programación en Python
 - 2.1. Sentencias
 - 2.2. Palabras reservadas , identificadores y variables
 - 2.3. Operadores
 - 2.4. Expresiones
 - 2.5. Estructuras de control
 - 2.5.1. Secuencial
 - 2.5.2. Selectivas
 - 2.5.2.1. If
 - 2.5.2.2. If-else
 - 2.5.2.3. If-elif
 - 2.5.3. Iterativas
 - 2.5.3.1. while
 - 2.5.3.2. for
 - 2.6. Subrutinas:

- 2.6.1. Procedimientos
- 2.6.2. Funciones
- 2.6.3. Parametros y argumentos
- 2.7. Funcion type()
- 2.8. Funcion dir()
- 2.9. Convertir de un tipo de dato a otro
- 2.10. Excepciones
 - 2.10.1. Try-except
- 2.11. Uso e instalacion de bibliotecas
 - 2.11.1. import
 - 2.11.2. from
 - 2.11.3. pip
 - 2.11.4. Ejemplos con biblioteca OS

Unidad 3 Funciones de cadena

Objetivo particular de la unidad: Conocer las funciones y operadores de cadena mas populares en Python.

Desarrollo:

- 3. Funciones de cadena
 - 3.1. Strings
 - 3.2. Inmutabilidad
 - 3.3. Slices
 - 3.4. Funcion len()
 - 3.5. Funcion Strip()
 - 3.6. Operador in
 - 3.7. Funcion find()
 - 3.8. Funcion Startswith()
 - 3.9. Format (%)
 - 3.10. Funcion capitalize()

Unidad 4 Archivos

Objetivo particular de la unidad: Aprender acerca del uso de archivos como recursos de persistencia; para poder extraer y almacenar informacion de ellos.

Desarrollo:

- 4. Archivos
 - 4.1. Abrir archivo
 - 4.2. Leer archivo
 - 4.3. Buscar en un archivo

Unidad 5 Estructuras de datos en Python

Objetivo particular de la unidad: Conocer las estructuras de datos mas usadas en python aplicado a ciencia de datos

Desarrollo:

- 5. Estructuras de datos en Python
 - 5.1. Lists

5.2. Dictionaries

5.3. Tuples

Unidad 6 Expresiones regulares

Objetivo particular de la unidad: Aprender a usar expresiones regulares para facilitar búsquedas dentro de cadenas.

Desarrollo:

6. Expresiones regulares

6.1. Caracteres especiales en expresiones regulares

6.2. Funcion search()

6.3. Funcion findall()

Unidad 7 Sockets y servicios web

Objetivo particular de la unidad: Conocer como obtener y usar informacion de la web

Desarrollo:

7. Sockets y servicios web

7.1. Sockets

7.2. Parsing

7.2.1. XML

7.2.2. JSON

7.3. Servicios web

7.3.1. RESTful API

7.3.2. Seguridad

Unidad 8 Bases de datos SQL

Objetivo particular de la unidad: Aprender acerca del uso de bases de datos SQL como recursos de persistencia; para poder extraer y almacenar informacion de ellas.

Desarrollo:

8. Bases de datos SQL

8.1.1. DDL

8.1.1.1. create

8.1.1.2. drop

8.1.1.3. alter

8.1.2. DML

8.1.2.1. select

8.1.2.2. insert

8.1.2.3. delete

8.1.2.4. update

8.1.2.5. upsert

8.1.2.6. Joins

8.2. Conexión a bases de datos SQL con Python

8.2.1. connect()

8.2.2. cursor()

8.2.3. execute()

- 8.2.4. close()
- 8.2.5. commit() y rollback()
- 8.2.6. executemany()

Unidad 9 Ambientes de desarrollo, depuración y pruebas de código

Objetivo particular de la unidad: Aprender como depurar y hacer pruebas de software a código escrito en python para optimizar y remover errores del software

9. Ambientes de desarrollo, depuración y pruebas de código

- 9.1. IDEs
 - 9.1.1. PyCharm
 - 9.1.2. Jupyter
- 9.2. Pruebas de software con PyTest

Unidad 10 Gestión de repositorios de código para control de versiones y desarrollo colaborativo con Git y GitHub

Objetivo particular de la unidad: Aprender a usar el protocolo git para control de versiones y conocer los clientes mas usados.

Desarrollo:

- 10. Gestión de repositorios de código para control de versiones y desarrollo colaborativo con Git y Github
 - 10.1. Sistemas de control de versiones
 - 10.2. Protocolo Git
 - 10.2.1. Historia de Git
 - 10.2.2. clone
 - 10.2.3. remote
 - 10.2.4. fetch
 - 10.2.5. push
 - 10.3. Integración con GitHub
 - 10.4. Integración con GitKraken

14. Actividades Prácticas

Resolver ejercicios de programación en Python y gestión de repositorios con git.

15.- Metodología

Lectura previa, estudios de casos, discusión de temas, cursos en línea, participación de los alumnos.

16.- Evaluación

1) Ejercicios básicos de Python	60%
2) Ejercicios intermedios de Python	25%
3) Ejercicio de Git	15%
Total	100%

17.- Bibliografía

1. Severance, C. Python for everybody: Exploring data in Python3.
2. Lutz, M. Learning Python: Powerful Object-Oriented Programming.

Otros materiales

www.py4e.com

<https://www.python.org/dev/peps/pep-0249/>

jupyter.org/

<https://www.jetbrains.com/pycharm/download/>

<https://www.jetbrains.com/help/pycharm/part-1-debugging-python-code.html>

<http://pytest.org/en/latest/contents.html>

<https://lab.github.com>

<https://www.coursera.org/learn/version-control-with-git>

<https://www.linkedin.com/learning/learning-git-and-github>

<https://www.linkedin.com/learning/git-essential-training-the-basics>

www.linkedin.com/learning/git-for-teams

www.linkedin.com/learning/git-branches-merges-and-remotes

<https://www.gitkraken.com/learn-git>

18.- Perfil del profesor

Profesor con nivel mínimo de maestría, de reconocido prestigio, con amplia experiencia en la docencia, programación y análisis de datos.

19.- Nombre de los profesores que imparten la materia

Fernando Carrillo Castro

20.- Lugar y fecha de su aprobación (incluyendo la última actualización)

Zapopan, Jal. 13 de enero de 2020

21.- Instancias que aprobaron el programa (Junta Académica y/ó Coordinación del programa)

Junta Académica de la MCD y el Departamento de Sistemas de Información.