

DIPLOMATURA EN SOFTWARE LIBRE

Curso: Algoritmos y Programación I Python

Planificación inicial y Temario detallado de estudio – Versión Preliminar tentativa

Institución: Universidad del Este - La Plata - Buenos Aires – Argentina

Período: 1º CUATRIMESTRE de 2015

Docente a cargo: Mg. Lic. Mariano Reingart

Horas semanales: 8

PRESENTACIÓN GENERAL

Python es un lenguaje de programación multipropósito publicado inicialmente en 1991 por Guido van Rossum en el CWI (Centro para las Matemáticas y Ciencias de Computación en los Países Bajos). Durante su estancia en CNRI (Corporación para Iniciativas de Investigación Nacional en EEUU), van Rossum lanzó el proyecto CP4E (Programación de computadoras para todos) auspiciado por DARPA (Agencia de Proyectos de Investigación Avanzada de la Defensa de EEUU), con el fin de hacer la programación más accesible.

Python tuvo un papel crucial en este proceso: era idóneo debido a su orientación hacia una sintaxis limpia y clara.

Python también es reconocido en el ámbito de aplicaciones científicas, desarrollo multimedia y juegos, aplicaciones web, entre otros. Principalmente por su estructura didáctica simple y clara es utilizado para la enseñanza secundaria, terciaria y universitaria alrededor del mundo. Su estructura dinámica e intérprete interactivo permite a los alumnos enfocarse rápidamente en los problemas y algoritmos a desarrollar, dejando de lado cuestiones triviales de sintaxis, compilación, etc. Esto es apoyado por la existencia de herramientas especialmente diseñadas con Python para el ámbito educativo (pythoncard/gui2py y web2py), que serán utilizados en el presente proyecto.

A su vez, Python es uno de los 10 lenguajes más utilizados a nivel mundial y forma parte de una fórmula ventajosa para la productividad, calidad y mantenibilidad del software en muchas compañías e instituciones alrededor del mundo, incluyendo Google y YouTube como algunos de los referentes más importantes.

OBJETIVOS

Este curso tiene como objetivo fundamental brindar los conocimientos, habilidades y el entrenamiento inicial para que el alumno pueda desarrollar soluciones mediante la programación utilizando estructuras de datos y control. Esto implica:

- Facilitar a los alumnos los conceptos teóricos necesarios para comprender la importancia de la programación de aplicaciones dinámicas en la actualidad.
- Facilitar el desarrollo de técnicas orientadas a la detección y depuración de errores de programación
- Facilitar el entendimiento, el desarrollo y la comprensión de la ejecución de programas de baja complejidad.

Por ello se espera que el alumno, al aprobar el curso, pueda:

- Reconocer el concepto de algoritmo.
- Entender las diferentes estructuras de datos y de control
- Diseñar algoritmos de baja complejidad.

- Caracterizar el concepto de programa
- Codificar los algoritmos en un lenguaje de programación estructurado y orientado a objetos como Python.

UNIDADES DIDÁCTICAS

Eje temático central: lenguaje de programación Python.

Unidad 1. Conceptos básicos

Computadoras y programas. Instrucciones (Python). Devolver resultados. El ciclo definido. Ayuda desde el intérprete. Construir programas y módulos. Estructura de los programas en Python. Estado y computación. Depuración de programas.

Unidad 2. Programas sencillos

Construcción de programas. Un ejemplo sencillo. Piezas de un programa Python. Nombres. Expresiones. Tipos de datos. Instrucciones. Ciclos definidos. Guía para el diseño

Unidad 3. Funciones

Documentación de funciones. Imprimir vs Devolver. Uso de funciones en un programa. Resultados de las funciones. Un ejemplo completo. Devolución múltiples resultados

Unidad 4. Decisiones

Expresiones booleanas. Expresiones de comparación. Operadores lógicos. Comparaciones simples. Múltiples decisiones consecutivas.

Unidad 5. Ciclos

Ciclos indefinidos. Ciclo interactivo. Ciclo con centinela. Cómo romper un ciclo.

Unidad 6. Cadenas de caracteres

Operaciones con cadenas. Segmentos de cadenas. Inmutabilidad. Procesamiento sencillo de cadenas.

Unidad 7. Tuplas y listas

Tuplas. Elementos. Inmutabilidad. Longitud. Empaquetado. Listas. Longitud. Elementos y segmentos. Mutabilidad. Búsqueda. Ordenamiento. Listas y cadenas.

Unidad 8. Algoritmos de búsqueda

El problema de la búsqueda. Búsqueda lineal. Búsqueda binaria.

Unidad 9. Diccionarios

Concepto. Utilización de diccionarios en Python. Usos.

Unidad 10. Contratos y mutabilidad

Pre y post condiciones. Aseveraciones. Invariantes de ciclo. Mutabilidad e Inmutabilidad.

Unidad 11. Manejo de Archivos

Apertura y cierre de archivos. Procesamiento. Escritura y Lectura. Manipulación binaria.

Persistencia de datos (CSV y binarios). Directorios

Unidad 12. Manejo de Errores y Excepciones

Errores. Excepciones. Manejo. Procesamiento y propagación. Acceso a la información de contexto. Validaciones.

Unidad 13. Procesamiento de archivos

Corte de control. Apareo.

Unidad 14. Objetos

Tipos. Concepto. Definición de nuevos tipos. Métodos especiales. Clases complejas.

Comparación. Ordenamiento.

Unidad 15. Polimorfismo, Herencia y Delegación

Polimorfismo. Interfaz. Redefinición de métodos. Ejemplo. Herencia. Delegación.

Unidad 16. Listas enlazadas

Clase sencilla. Tipos Abstractos de Datos. Invariantes de Objetos. Otras listas enlazadas. Iteradores.

Unidad 17. Pilas y colas

Pilas. Pilas representadas por listas. Uso. Colas. Colas implementadas sobre listas.

Unidad 18. Modelo de ejecución de funciones y recursividad

La pila de ejecución de funciones. Pasaje de parámetros. Devolución de resultados.

Recursión. Iteración. Ejemplos. Limitaciones.

Unidad 19. Ordenar listas

Ordenamiento por selección. Ordenamiento por inserción.

Unidad 20. Algunos ordenamientos recursivos

Ordenamiento por mezcla o Merge Sort. Ordenamiento rápido o Quick Sort. Versión mejoradas. Costo.

METODOLOGÍA

Cada alumno trabajará desde su computadora individual comunicados vía Internet por el campus virtual, enfocados en una guía de ejercicios para cada unidad.

El profesor asumirá un método de enseñanza directa ya que conducirá y dictará las pautas a seguir, realizando una exposición del tema a desarrollar (con demostraciones de las actividades prácticas o presentación de temas teóricos), para que luego los alumnos puedan reproducir y analizar los nuevos contenidos a aprender.

Se busca un aprendizaje significativo, favoreciendo que los alumnos relacionen con mayor facilidad los conocimientos aprendidos a lo largo de la diplomatura. Este tipo de aprendizaje es el que hará efectiva la comprensión de los temas, donde habrá un proceso de elaboración de conocimiento de una manera integral.

EVALUACIÓN

Las evaluaciones serán formativas con base constructivista (evitando la mera memorización mecánica o repetitiva), de carácter procesual (teniendo en cuenta los procedimientos realizados y no solo el resultado final), con contenidos teóricos y prácticos, llevadas a cabo principalmente en el campus virtual.

El alumno deberá poder contestar preguntas teóricas sobre los contenidos vistos en el curso y desarrollar ejercicios prácticos usando las herramientas utilizadas durante la cursada.

Se evaluará de manera continuada, bajo modelos de evaluación tradicional (con varios exámenes parciales domiciliarios por cuatrimestre: actividades -trabajos prácticos- a entregar por el campus virtual) y enfoques alternativos actualizados medida por tecnología (observaciones de la participación y colaboración activa con criterios concretos). Se apunta a una evaluación sumativa de todos los conceptos, que refiera a los logros y rendimiento del aprendizaje de los alumnos, debiendo poner de manifiesto la internalización de los conceptos abordados.

Cada evaluación parcial será escrita por computadora (con elementos multimedia), en la que el alumno desarrollará los contenidos teórico-prácticos vistos en el curso en base a una guía de examen facilitada por el profesor. El alumno deberá elaborar y entregar en tiempo y forma todas las actividades obligatorias que proponga el equipo docente.

Actividades:

Ver ejercitación en la bibliografía (una actividad a entregar por unidad)

Calificación:

La calificación será porcentual (0 a 100), siendo 70 el valor mínimo para aprobar los exámenes, actividades y ejercicios. No se contemplan recuperatorios, por lo que el alumno deberá aprobar al menos 2/3 de las actividades propuestas por el docente. Para acreditar los conocimientos, el alumno deberá aprobar los exámenes parciales. Una vez que el alumno

haya cumplido con la aprobación de los exámenes, se entregará al alumno un certificado emitido por la universidad donde conste que el alumno ha aprobado el curso.

EXPERIENCIA DOCENTE

Mariano Reingart es Licenciado en Sistemas, Magíster en Software Libre (UOC) y actualmente finalizando el Profesorado en Disciplinas Industriales (UTN-INSPT). Es docente en el Instituto Superior Blaise Pascal desde 2009 en materias de 2º y 3º Año (actualmente «Bases de Datos», «Sistemas Operativos», «Interconectividad» y «Práctica Profesional») de las carreras terciarias «Tecnicatura superior en Análisis de Sistemas / Redes Informáticas». Ha trabajado como Analista-Programador Freelance (en varias empresas del sector y actualmente en un emprendimiento propio). En el área del software libre, es miembro de varias asociaciones y grupos de usuarios/desarrolladores, con activa participación en varios proyectos. En especial, es miembro de la Python Software Foundation, habiendo disertado en varias conferencias y colaborado en múltiples proyectos de software libre que utilizan este lenguaje de programación.

REFERENCIAS Y ANTECEDENTES

Materia “Algoritmos y Programación I”

Facultad de Ingeniería – Universidad de Buenos Aires – Catedra Wachenchauzer
<https://sites.google.com/site/fiuba7540rw/>

BIBLIOGRAFÍA GENERAL

Algoritmos y Programación I: Aprendiendo a programar usando Python como herramienta

Rosita Wachenchauzer , Margarita Manterola, Maximiliano Curia, Marcos Medrano, Nicolás Paez. Buenos Aires, 2012
https://sites.google.com/site/fiuba7540rw/home/apunte_7540.pdf