

Propuesta formativa (v4 actualizada a 06-04-2020)

Introducción a la Programación con Python

y su aplicación en NLP (Natural Language Processing)



Descripción del curso

El curso Introducción a la Programación con Python y su aplicación en NLP introduce al alumno en los conceptos de programación básicos, haciendo especial hincapié en la presentación de las técnicas algorítmicas esenciales y la utilización de diferentes paradigmas, lenguajes y entornos de programación para la construcción de aplicaciones informáticas, aplicando metodologías de manera rigurosa que proporcionen al alumno buenos hábitos en el desarrollo de software.

El curso se centra en la resolución de problemas, construyendo programas que resuelvan problemas sencillos. Para ello parte del diseño del algoritmo, utilizando estructuras modulares y distintos métodos de programación, y alcanza tanto la implementación del mismo en un lenguaje de programación estructurada, utilizando estructuras de control fundamentales de la codificación y tipos de datos básicos, como su posterior compilación, depuración y documentación.

En enfoque hacia el procesamiento de lenguaje natural permitirá al alumno empezar a trabajar de manera directa sobre la interacción entre el ordenador y el lenguaje humano, analizando y categorizando textos digitalizados.

Además, proporciona los aspectos básicos que comprenden las principales áreas de la informática: representación, gestión y almacenamiento de datos, programas, compiladores y fundamentos de la programación, tipos de sistemas informáticos y su estructura hardware, sistemas operativos y principios de conectividad y de las redes de ordenadores.

Conocimientos previos

Se recomienda disponer de destrezas en el manejo del ordenador personal.

También se recomienda disponer de los conocimientos básicos de álgebra y aritmética.

Objetivo

El objetivo principal del curso consiste en presentar a los estudiantes los principios de la resolución sistemática de problemas a través de el diseño de algoritmos y codificación de una solución, y las reglas básicas de la computación. El lenguaje que se utilizará para transmitir los contenidos es **Python**. De este modo, el alumno se empieza a familiarizar con un lenguaje de propósito general.

Dar a entender y aplicar los principios fundamentales y conceptos de la programación de ordenadores, incluyendo la abstracción, lógica, algoritmos y representación de datos.

Introducir al alumno en el procesamiento del lenguaje natural, analizar, procesar, clasificar y categorizar textos, extraer información y realizar un análisis significativo de la información.

Aplicar el pensamiento computacional para la resolución de problemas, entendiendo los principios fundamentales sobre la resolución de problemas a través del uso de los ordenadores.

Proporcionar una visión global de la Informática.

Los cambios que se avecinan a nivel tecnológico son de tal envergadura que cuesta anticipar sus consecuencias e impacto. Conocer las tendencias tecnológicas, manejar las tecnologías actuales y estar preparados para las futuras, así como poder anticipar y reflexionar acerca del impacto y consecuencias que estos cambios traerán a la vida de las personas.

Entender el para qué de la Informática, cuál es el fin último que persigue y el bien que aporta a la sociedad.

Reflexionar acerca del rol que cada uno quiere jugar en la sociedad del futuro y empezar a diseñar su proyecto profesional.

Contenidos Bloque Programación

[illegible]

Contenidos Bloque Programación

Unidad 1. Introducción a la programación

- ¿Qué es la Programación?
- Lenguajes de Programación
- Paradigmas de Programación
- Abstracción
- Algoritmos + Estructuras de datos = Programas

Unidad 2. Tipos y Operaciones

- Características del Lenguaje.
- Estructura de un programa.
- Diferentes etapas de la vida de un programa.
- Entornos de desarrollo.
- Introducción al entorno de trabajo REPL y Colaboratory.
- Crear un nuevo entorno Python.
- Distribuir Paquetes.
- Instalación y Configuración de Python.
- Introducción al Interprete de Python.
- Como lanzar programas de Python.
- Tipos Numéricos.
- Tipado Dinámico en Python.
- Operadores Aritméticos / Operador Suma y Resta.
- Operador de Negación / Operador Exponente y División.
- Ficheros: Manejo y Formatos.
- Otros: Tipos de Python.

Unidad 3. Sintaxis, Sentencias y Funciones

- Revisión de la Jerarquía de Python.
- Ejemplo sencillo: Bucles Interactivos.
- Asignación, Expresiones e Imprimir por Pantalla.
- Sentencia condicional "If".
- Bucles "For" y "While".
- Definición, Llamadas y Polimorfismo de Funciones en Python.
- Ámbito de los Elementos de Python: variables locales y globales.
- Argumentos de Funciones.
- Herramientas de Programación Funcional.
- Representación de algoritmos: diagramas de flujo y pseudocódigo.
- Comprobación del Timing.

Unidad 4. Módulos y Paquetes

- Introducción a los módulos.
- Importación y Uso de módulos.
- Espacio de nombres de los módulos.
- Paquetes de módulos.
- Espacio de nombres de paquetes de Python.
- Temas Avanzados sobre Módulos.
- Modos de Uso Mixtos.

Contenidos Bloque Programación

Unidad 5. Programación Orientada a Objetos

- Introducción a las Clases y Objetos.
- Elementos y Características de la POO.
- Construcción de Objetos.
- Programación Básica de Clases.
- Personalizar el formateado de Cadenas.
- Propiedades de las Clases.
- Instancias, Métodos y Atributos.
- Herencia. Clases de Estilo Nuevo.
- Invocar un método en la Clase Padre.
- Accediendo a los Métodos y Propiedades de un Objeto.
- Métodos y Clases Estáticas.

Unidad 6. Excepciones, Estructuras de datos y Algoritmos

- Detalles de Programación con Excepciones.
- Consejo y Ejemplos de Diseño de código.
- Guardar los últimos N elementos.
- Encontrar los N Mayores o Menores Elementos.
- Implementar una Cola con Prioridad.
- Mapear Nombres a Elementos de Secuencias.
- Transformar y Reducir Datos al mismo tiempo.
- Combinar varios diccionarios en uno.

Unidad 7. Cadenas, Texto, Números, Fechas y Horas

- Buscar patrones de texto.
- Búsqueda y sustitución de texto.
- Interpolación de variables dentro de Cadenas.
- Re-formatear texto a un número fijo de columnas.
- Manejar entidades HTML y XML en un texto.
- Convertir Texto a Símbolos.
- Escribir un Analizador Sintáctico Recursivo.
- Formatear número para mostrar.
- Cálculos con fracciones.
- Números aleatorios.
- Conversiones de tiempo.
- Saber el rango de fechas de un mes.
- Convertir cadenas a Datetimes.

Contenidos Bloque Programación

Unidad 8. Iteradores y generadores. Ficheros de Entrada / Salida

- Implementar el protocolo del Iterador.
- Iteración en orden inverso.
- Tomar porciones(Slices) de generadores.
- Crear pipelines de procesamiento de datos.
- Sustituir bucles While infinitos por un iterador.
- Manipular nombres y rutas de ficheros.
- Imprimir nombres de ficheros corruptos.
- Comunicación con puerto serie.
- Serializar objetos.

Unidad 9. Codificación y procesamiento de datos

- Leer y escribir datos CSV.
- Leer y escribir datos JSON.
- Agregar datos y realizar estadísticas.

Unidad 10. Funciones avanzadas

- Devolver múltiples valores en una función.
- Definir funciones con argumentos por defecto.
- Definir funciones anónimas.
- Sustituir clases con un solo método por funciones.
- Acceder a variables definidas dentro de una clausura.

Unidad 11. Análisis de Datos

- Arrays Multidimensionales.
- Visualización de Datos.
- Uso de las librerías Pandas y Numpy
- Representación de datos con Matplotlib, Seaborn y HighCharts.

Unidad 12. Contenidos Específicos del módulo NLP

- Conceptos básicos sobre el NLP o procesamiento del lenguaje natural
- Introducción a la librería NLTK
- Accesos de datos
- Tipos de entradas de datos
- Análisis de textos
- Estadísticas
- Modelos de representación de resultados. Wordcloud
- Categorización
- Clasificación
- Extracción de información
- Análisis significativo

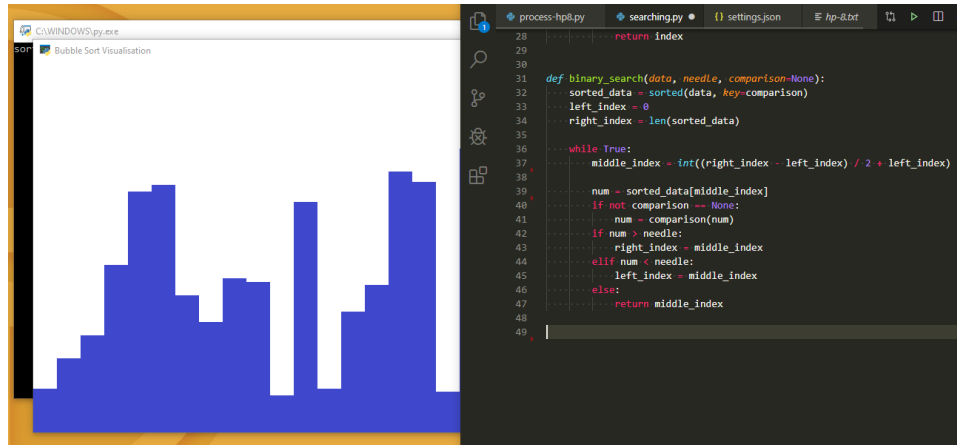
Contenidos Bloque Programación

Unidad 13. Comprobación de Tipos

- Sistemas de Tipado.
- Anotaciones de Función.
- Anotaciones de Variable.
- Comentarios de Tipo.
- Comprobación Estática de Tipos.

Unidad 14. Buenas Prácticas de Programación en Python

- Guía de Estilo.
- Documentación de Código.
- Construcciones Idiomáticas.



Acciones formativas

La metodología seguida en este curso está dirigida a conseguir un aprendizaje significativo por parte del alumno de los conceptos y técnicas fundamentales de la materia. Por ese motivo se combinan lecciones expositivas con clases prácticas y presentación de trabajos, de manera que se favorezca la participación del alumno y la interacción alumno-profesor como vía para fomentar el aprendizaje colaborativo y la capacidad de autoaprendizaje, todo ello mediante estrategias de resolución de problemas y metodologías de intervención. Las actividades no presenciales, serán supervisadas por el profesor en clases y tutorías, tanto individuales como de grupo, estando encaminadas a favorecer el aprendizaje autónomo y colaborativo en un entorno creado con las herramientas de desarrollo y trabajo de G-Suite y Colaboratory de Google.

Las actividades de carácter no presencial previstas incluyen el estudio individual, que permitirá trabajar en la fijación de los conceptos teóricos abordados en las clases expositivas correspondientes a todas las materias del módulo y adquirir la destreza práctica que se persigue con las clases prácticas, que aplicarán el aprendizaje por descubrimiento. Para el desarrollo de las competencias y habilidades en este curso son igualmente importantes los trabajos individuales. En algunos casos, el alumno tendrá que realizar en clase la exposición de las principales conclusiones de su estudio o trabajo, lo que permitirá el intercambio de conocimientos y experiencias entre alumnos.

Para garantizar la comprensión de los algoritmos explicados, se plantearán ejercicios prácticos de realización individual y autónoma dentro y fuera del aula, donde los alumnos tendrán que mostrar la secuencia ordenada de pasos que supone la ejecución de cada algoritmo.

Además, se pretende contribuir con este curso al desarrollo de competencias básicas esenciales para un futuro.

Distribución de tiempos de trabajo

	Periodo	Semanas	Sesiones Semana	Duración Sesión	Tutoría presencial después de cada sesión presencial	Horas de tutoría no presencial por semana	Dedicación personal de alumno por semana	Total horas formación presencial	Total horas tutoría presencial	Total horas tutoría no presencial	Total horas dedicación personal del alumno
Clases	Del 12 de abril al 30 de mayo	7	3	2	0,5	3	6	42	10,5	21	42
Proyecto personal	Del 1 de junio al 28 de junio	4	4		1	4	15		16	16	60
Corrección y evaluación del proyecto personal	Del 29 de junio al 5 de julio	1									
								42	26,5	37	102

Competencias

Competencias básicas:

- Que los estudiantes sepan aplicar sus conocimientos a su trabajo o vocación de una forma profesional y posean las competencias que suelen demostrarse por medio de la elaboración y defensa de argumentos y la resolución de problemas dentro de su área de estudio

Competencias generales:

- Conocimiento de las materias básicas y tecnologías, que capaciten para el aprendizaje y desarrollo de nuevos métodos y tecnologías, así como las que les doten de una gran versatilidad para adaptarse a nuevas situaciones.

Competencias específicas:

- Capacidad para comprender y dominar los conceptos básicos de matemática discreta, lógica, algorítmica y complejidad computacional y procesamiento de lenguaje natural y su aplicación para la resolución de problemas.
- Conocimientos básicos sobre el uso y programación de los ordenadores, sistemas operativos, bases de datos.
- Conocimiento de la estructura, organización, funcionamiento e interconexión de los sistemas informáticos, los fundamentos de su programación, y su aplicación para la resolución de problemas de aplicación práctica y cotidiana.
- Capacidad para evaluar la complejidad computacional de un problema, conocer estrategias algorítmicas que puedan conducir a su resolución y recomendar, desarrollar e implementar aquella que garantice el mejor rendimiento de acuerdo con los requisitos establecidos.

Resultados del aprendizaje

- Reconocer y usar, en una explicación la terminología, tanto hardware como software, involucradas en la realización, gestión y explotación de un sistema informático, relacionada con los principales ámbitos de conocimiento de la informática.
- Diseñar y desarrollar parcialmente una solución informática que de respuesta a una necesidad de algún colectivo de personas.
- Analiza e interpreta el problema a resolver y elabora algoritmos utilizando las estructuras adecuadas de control y de datos.
- Depura los errores detectados y analiza las posibles alternativas de resolución del problema.
- Diseña e implementa algoritmos que resuelvan problemas sencillos utilizando tipos de datos elementales y estructuras de control básicas.
- Analiza y selecciona la iteración adecuada para cada caso.
- Implementa un pseudocódigo detallado, estructurado y conciso sobre un ejemplo dado. Detecta el uso de funciones recursivas en los casos que proceda.
- Demuestra destreza en el uso de las herramientas necesarias para el correcto desarrollo software incluyendo librerías, control de versiones, documentación, etc.
- Analiza la complejidad de problemas reales que se dan en el ámbito del pensamiento computacional.
- Analiza la complejidad de algoritmos, siendo capaz de expresar esta en el caso mejor, promedio y peor.
- Utilizar el pensamiento computacional para la resolución de problemas distinguiendo las etapas de análisis, descomposición, reconocimiento de patrones, abstracción y algoritmo.
- Entiende los principios fundamentales del procesamiento del lenguaje natural.
- Distingue las distintas acciones que resultan de la utilización de herramientas de procesamiento del lenguaje natural.
- Representa gráficamente e interpreta datos de distintas fuentes.

Sistema de evaluación del aprendizaje

El sistema de evaluación contempla tres tipos de pruebas:

- Practicas y otros trabajos relacionados con el curso: presenta un peso del 40% en la nota final.
- Trabajo final: presenta un peso del 40% en la nota final.
- Participación en clase e implicación en el curso: presenta un peso del 20% en la nota final, siendo requisito imprescindible haber asistido como mínimo al 80% de las sesiones. En caso contrario este tipo de prueba se calificará con 0 puntos.

El peso por objetivos de la evaluación se distribuye de la siguiente manera:

- Recordar, seleccionar y comunicar el conocimiento y entendimiento de la computación: 40%
- Aplicar el conocimiento, entendiendo y demostrando destrezas para resolver problemas informáticos y de programación: 40%
- Analizar, evaluar y emitir juicios razonables, presentando conclusiones: 20%

Requerimientos técnicos

Hardware:

- Ordenador de sobremesa o portátil con conexión a internet.

Software:

- Navegador, preferiblemente Google Chrome.
- Videollamadas por Zoom.
- Acceso a Google Classroom, Hangouts y Meet.
- Acceso a REPL.it
- Opcionalmente los alumnos que lo deseen realizar su proyecto personal con Visual Studio Code y el interprete de Python.org
- Tech Talents podrá crear una cuenta de estudiante de Google G-Suite a todos aquellos alumnos que los necesiten sin coste adicional alguno.

Propuesta formativa

- **Lugar:** Formación a distancia en modalidad presencial virtual.
- **Requerimientos técnicos:** BYOD, adicionalmente Tech Talents pondrá a disposición de los alumnos los equipos necesarios.
- **Grupo máximo recomendado:** 25 alumnos.

Clases presenciales virtuales

- **Sesiones por semana:** 3, Lunes, Miércoles y Viernes
- **Duración de la sesión lectiva:** 2,0 horas.
- **Disponibilidad del profesor después de cada sesión lectiva para tutoría:** 0,5 hora.
- **Horas semanales adicionales de seguimiento y tutorías remotas:** 3 horas.
- **Duración de clases presenciales:** del 12 de abril al 30 de mayo de 2020. Total: 7 semanas, 3 clases por semana de 2 horas cada clase.

Proyecto personal

- **Estimación total de horas de trabajo semanales por parte del alumno:** 15 horas.
- **Disponibilidad del profesor para tutorías de grupales:** 1 hora diaria, 4 días cada semana.
- **Horas semanales adicionales de seguimiento y tutorías remotas:** 4 horas.
- **Duración del proyecto personal:** del 1 de junio al 28 de julio de 2020. Total: 4 semanas.
- **Plazo corrección y evaluación de proyectos:** 1 semana.

- El presente documento ha sido elaborado con exclusividad para Uría Menéndez por Tech Talents y está destinado únicamente para el efecto contemplado en el mismo y su destinatario.
- Queda prohibido cualquier uso no permitido por Tech Talents, así como su reproducción, difusión, publicidad no consentida, cesión, comunicación a terceros o a personal ajeno y externo a Uría Menéndez, así como su modificación, total o parcial, sin autorización expresa y previa por Tech Talents.
- La contravención de la confidencialidad dará derecho a la indemnización oportuna por daños y perjuicios así como los demás derechos inherentes a la misma.
- Estas obligaciones son aplicables a todo el personal de Uría Menéndez que tenga o haya podido tener acceso al documento.

Propuesta formativa (v4 actualizada a 06-04-2020)

Introducción a la Programación con Python y su aplicación en NLP (Natural Language Processing)