# Programación I

En el primer curso de programación siempre se da teoría básica de las ciencias de la computación acompañada de su implementación en algún lenguaje de programación. Se recomienda que Python sea el primer lenguaje de programación porque es fácil de aprender, requiere menos código para realizar tareas básicas, y porque permite fácilmente migrar a otros lenguajes de programación, pues sus metodologías pueden ser utilizadas en un amplio rango de aplicaciones.

Siguiendo el contenido del curso *MITx: 6.00.1x Introduction to Computer Science and Programming*, se recomienda que el primer curso de programación cubra los contenidos descritos a continuación, pues son esenciales para comprender cómo funciona cualquier otro lenguaje de programación.

1. Introducción:
   1. Pensamiento computacional
   2. Aspectos de los lenguajes de programación
   3. Arquitectura básica de la máquina
2. Elementos básicos de los programas:
   1. Tipos de lenguajes de programación
   2. Objetos, expresiones y operadores
   3. Instrumentos de cuerda
   4. *Scripts*
   5. Estructurar un programa
3. Algoritmos simples:
   1. Programas sencillos
   2. Iteración
   3. Entradas
   4. Ciclos *for* y *while*
4. Funciones:
   1. Sintaxis de funciones
   2. Abstracción por especificación
   3. Funciones y su espacio de trabajo
   4. Números de punto flotante
   5. Módulos
5. Recursividad:
   1. Recursividad
   2. Razonamiento inductivo
   3. Estrategia “Divide y conquistarás”
6. Objetos:
   1. Tipos estructurados y mutabilidad
   2. Tuplas
   3. Listas y mutabilidad
   4. Funciones como objetos
   5. Diccionarios
7. Depuración:
   1. Pruebas y depuración
   2. Prueba de caja negra
   3. Prueba de caja de vidrio
   4. Pruebas de integración y pruebas unitarias
8. Aserciones y Excepciones
   1. Afirmaciones
   2. Excepciones
9. Eficiencia y órdenes de crecimiento:
   1. Complejidad
   2. Cómo medir la complejidad
   3. Notación asintótica
10. Memoria y búsqueda:
    1. Almacenamiento de memoria
    2. *Indirection* (indirección)
    3. Métodos de búsqueda y clasificación
    4. *Hashing*
11. Clases:
    1. Definición de clases
    2. Clases versus instancias
    3. Métodos
    4. Excepciones
12. Programación orientada a objetos y herencia:
    1. Herencia
    2. Programación orientada a objetos
13. Árboles:
    1. Árboles de decisión
    2. Búsqueda por profunidad
    3. Búsqueda por ramificaciones

# Programación II

Ahora con conocimientos básicos de ciencias de la computación y con un lenguaje de programación de referencia, sería conveniente que se haga aplicación de otros lenguajes más orientados a la “ciencia de datos”, dentro de los que se puede mencionar como ejemplo: Python (con sus módulos de numpy, scipy y matplotlib), MATLAB y R.

Para este curso se tomará como referencia los cursos *MITx: 6.00.2x Introduction to Computational Thinking and Data Science,* y los cursos de ciencias de datos ofrecidos por Microsoft en edX: *Introduction to R for Data Science* e *Introduction to Python for Data Science*. La idea es que aprendan dos lenguajes más de programación y al mismo tiempo hagan uso de ellos en aplicaciones de simulación, y serviría como introducción al curso final, propio de las simulaciones de modelos en R.

1. Python: numpy (librería numérica)
2. Python: matplotlib
3. Introducción a R:
   1. Vectores
   2. Matrices
   3. *Data frames*
   4. Listas
   5. Creación de gráficas
   6. Scripts y algoritmos
4. Introducción a MATLAB:
   1. Operaciones con vectores y matrices
   2. Scripts y algoritmos
   3. Creación de gráficas
5. Simulaciones y caminatas aleatorias
6. Probabilidad y distribuciones
7. Creación de histogramas
8. Simulaciones de Monte Carlo
9. Ajuste de curvas (modelos lineales)