IIC1103 - Sección 4

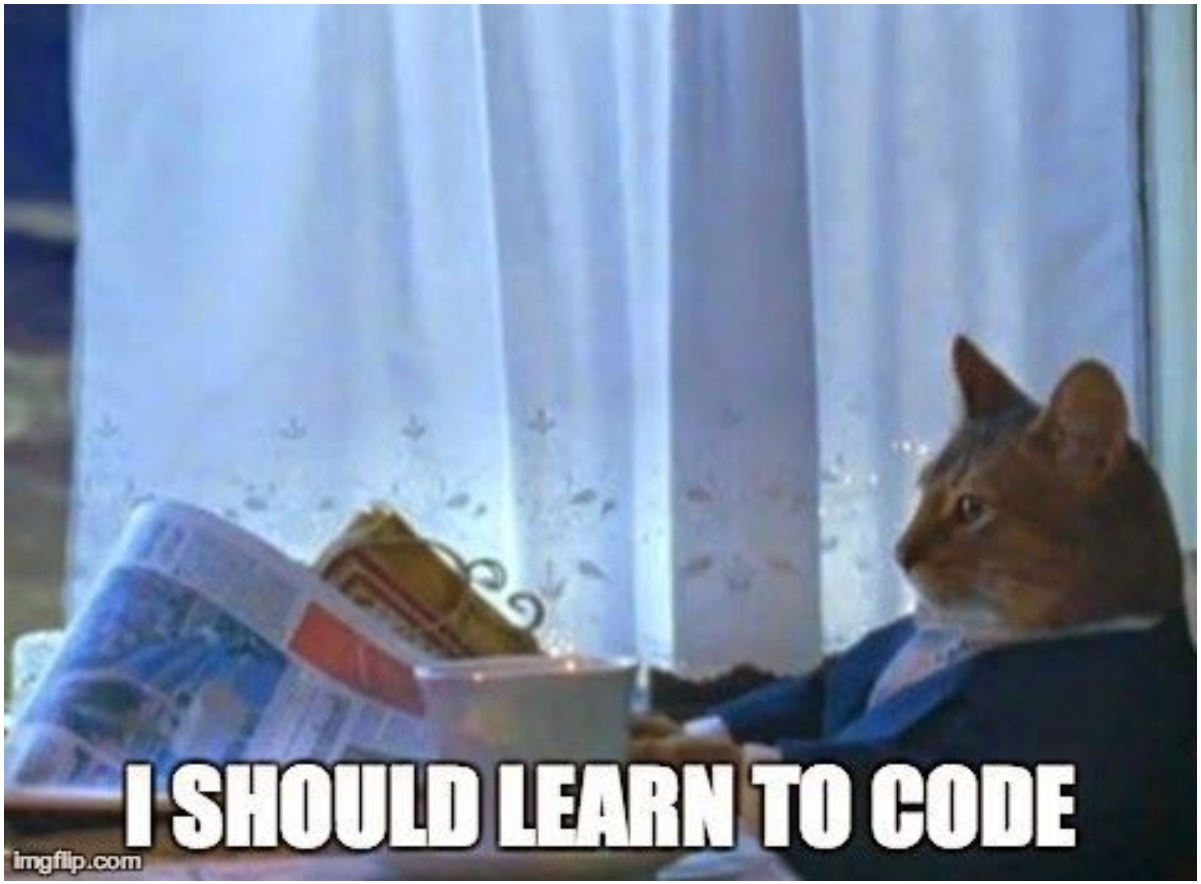
Clase 1: Bienvenida y reglas

Prof. Francisca Cattan

Universidad Católica de Chile

6 de marzo, 2023

¡Bienvenidas y bienvenidos a introducción a la programación!



¿Quién soy yo?

- Francisca Cattan
  fpcattan@uc.cl
- Arquitecta y M.Arch. UC, graduada en 2017.
- Realizando un doctorado en Inteligencia Artificial aplicada a Robótica Cognitiva en la UC.
- Ex desarrolladora de Inteligencia Artificial en Zippedi.
- Soy extra nerd y se van a dar cuenta.



¿Quién es su ayudante de sección?

- Martín Atria
  martin.atria@uc.cl
- Major Robótica.
- Delegado de la gen 21.
- Soy Jefe scout y mi hobbie es hacer malabares.
- Fui top 1 del mundo en Quiz Up en la categoria How I Met Your Mother



De qué se trata este curso

Este curso es una introducción a los fundamentos de la programación, en donde entregamos los conceptos esenciales que permitan a los estudiantes poder crear algoritmos que resuelvan problemas.

De qué se trata este curso

Este curso es una introducción a los fundamentos de la programación, en donde entregamos los conceptos esenciales que permitan a los estudiantes poder crear algoritmos que resuelvan problemas. El objetivo es implementar en un lenguaje de programación mediante ciertos **comandos**, usando una **sintaxis** y un sistema de **reglas**.

No requiere ningún conocimiento previo en programación.

Objetivo del curso

Desarrollar las competencias básicas necesarias para resolver problemas de diversa índole mediante la programación de computadores. Se espera que los estudiantes aprendan a:

Objetivo del curso

Desarrollar las competencias básicas necesarias para resolver problemas de diversa índole mediante la programación de computadores. Se espera que los estudiantes aprendan a:

1. Elaborar modelos y algoritmos que permitan resolver estos problemas usando el enfoque de orientación a objetos.

Objetivo del curso

Desarrollar las competencias básicas necesarias para resolver problemas de diversa índole mediante la programación de computadores. Se espera que los estudiantes aprendan a:

1. Elaborar modelos y algoritmos que permitan resolver estos problemas usando el enfoque de orientación a objetos.
2. Escribir programas que implementen estos algoritmos utilizando un lenguaje de programación.

- El lenguaje de programación a utilizar será Python (versión 3).

Tabla de contenidos

1. Introducción a los algoritmos.
2. Variables, datos y operaciones.
3. Control de flujo — Condiciones.
4. Control de flujo — Ciclos.
5. Funciones.
6. Recursión.
7. Strings.
8. Listas.
9. Archivos.
10. Programación orientada a objetos — Clases.
11. Ordenamiento.

¿Por qué aprender a programar?

¿Por qué aprender a programar?

- ¿Debería darme el tiempo de aprender lo básico de programar?
  - ¡SI!

¿Por qué aprender a programar?

- ¿Debería darme el tiempo de aprender lo básico de programar?
  - ¡SI!
- ¿Va a ser difícil aprender a programar?
  - ¡NO!

¿Por qué aprender a programar?

- ¿Debería darme el tiempo de aprender lo básico de programar?
  - ¡SI!
- ¿Va a ser difícil aprender a programar?
  - ¡NO!
- ¿Es como andar en bicicleta?
  - Si te esfuerzas por aprender correctamente, ¡SI!

¿Por qué aprender a programar?

- ¿Debería darme el tiempo de aprender lo básico de programar?
  - ¡SI!
- ¿Va a ser difícil aprender a programar?
  - ¡NO!
- ¿Es como andar en bicicleta?
  - Si te esfuerzas por aprender correctamente, ¡SI!
- ¿La programación está restringida a una sola aplicación?
  - ¡NO! Casi todo lo que nos rodea hoy en día tiene una base o inicio en la programación.

¿Por qué aprender a programar?

- La programación es una habilidad muy buscada en el mundo profesional.
- Vamos a aprender un lenguaje, pero aún más allá vamos a aprender un forma de pensar. Una forma de afrontar todo tipo de problemas.
- Saber sólo un poco te dará una enorme ventaja.
- Si les interesa la tecnología, debieran aprender a programar.
- No solo para graduados del major de computación – si quieren liderar una empresa de tecnología, tienen que saber qué pasa en los procesos y qué hacen sus desarrolladores.

¿Cómo se programa?

La verdad no tengo idea. Recibo ideas...

¿Cómo se programa?

Especificando una serie de pasos: un **algoritmo**.
Un algoritmo es una secuencia ordenada de instrucciones.

¿Cómo se programa?

Especificando una serie de pasos: un **algoritmo**.
Un algoritmo es una secuencia ordenada de instrucciones.

- ¿Cómo preparar un pie de limón?
  1. Preparar la masa.
  2. Preparar el relleno.
  3. Preparar el merengue.
  4. Apilar los ingredientes en orden.
  5. Ponerlo en el horno.
  6. Esperar.

¿Cómo se programa?

Especificando una serie de pasos: un **algoritmo**.
Un algoritmo es una secuencia ordenada de instrucciones.

- Algoritmo para llegar de San Joaquin a Plaza Italia

¿Cómo se programa?

Especificando una serie de pasos: un **algoritmo**.
Un algoritmo es una secuencia ordenada de instrucciones.

- Algoritmo para llegar de San Joaquin a Plaza Italia
  1. Salir de San Joaquin
  2. Tomar el metro.

  3. Bajarse en Baquedano.

¿Cómo se programa?

Especificando una serie de pasos: un **algoritmo**.

Un algoritmo es una secuencia ordenada de instrucciones.

- Algoritmo para llegar de San Joaquin a Plaza Italia
  1. Salir de San Joaquin
  2. Tomar el metro.
     - 2.1 Sacar la tarjeta Bip!
     - 2.2 Revisar si tengo carga.
     - 2.3 Pagar.
     - 2.4 Esperar el tren.
     - 2.5 Subirse al tren.
  3. Bajarse en Baquedano.

¿Cómo se programa?

Especificando una serie de pasos: un **algoritmo**.

Un algoritmo es una secuencia ordenada de instrucciones.

- Algoritmo para llegar de San Joaquin a Plaza Italia
  1. Salir de San Joaquin
  2. Tomar el metro.
     - 2.1 Sacar la tarjeta Bip!
     - 2.2 Revisar si tengo carga.
     - 2.3 Pagar.
     - 2.4 Esperar el tren.
     - 2.5 Subirse al tren.
  3. Bajarse en Baquedano.
     - 3.1 Confirmar que se abran las puertas.
     - 3.2 Bajar del tren.
     - 3.3 Salir de la estación.


I'm bored now.

Material de trabajo

— Slides!

Las slides de las clases serán el material principal de teoría y aplicación. Estarán disponibles en Canvas al inicio de la semana correspondiente a cada cátedra.

Material de trabajo

— Slides!

Las slides de las clases serán el material principal de teoría y aplicación. Estarán disponibles en Canvas al inicio de la semana correspondiente a cada cátedra.

— Notebooks!!

Haremos ejercicios prácticos con código comentado. Este material es para ustedes y les sirve como método principal de ejercitación. En ellos cubriremos la mayoría de los casos clásicos y borde de cada elemento de la materia. Liberaré las soluciones al final del día.

Material de trabajo

— Slides!

Las slides de las clases serán el material principal de teoría y aplicación. Estarán disponibles en Canvas al inicio de la semana correspondiente a cada cátedra.

— Notebooks!!

Haremos ejercicios prácticos con código comentado. Este material es para ustedes y les sirve como método principal de ejercitación. En ellos cubriremos la mayoría de los casos clásicos y borde de cada elemento de la materia. Liberaré las soluciones al final del día.

— Sets!!!

Todas las secciones tienen acceso a una serie de "Sets de Ejercicios" llamadas amorosamente "Sets". Son 14 Sets diseñados para sacar el jugo a cada elemento de la materia. Realizarán estos en la plataforma Clearn y contarán como puntaje de participación en nuestra sección.


BRACE YOURSELVES
CLASS RULES ARE COMING

Reglas: Clases

Cátedra presencial: Lunes y miércoles módulo 2 – 10:00 a 11:20

Ayudantía presencial (SAP): Sala de Ayudantía Presencial en el campus San Joaquin. Habrá varios ayudantes y muchas salas para resolver dudas sobre sets de ejercicios, tareas y material de estudio. L-M-W-J-V módulos 5 y 6 – 15:30 a 18:20

Reglas: Ayudantes

Ayudantes SAP: Disponibles en el campus San Joaquin. No debieran contactarlos por otros medios ni fuera de horario. Resuelven todo tipo de dudas de ejercitación, materia, tareas e interrogaciones. Nos acompañarán durante las pruebas presenciales en sala.

Martín ayudante de cátedra: Vendrá un par de veces, pero principalmente responderá a su correo martin.atria@uc.cl Lo pueden molestar a cualquier hora (decente) y cualquier día. Resuelve dudas de materia, actividades, clases, y nota de participación. ¡No resuelve dudas de tareas ni interrogaciones!¡No revisa inasistencias!

Reglas: Evaluaciones

- 3 Tareas:
  - $T1$: 10 - 26 Abr. (16 días)
  - $T2$: 15 - 31 May. (16 días)
  - $T3$: 12 - 30 Jun. (18 días, 16 sin feriados)
- 2 Interrogaciones:
  - $I1$: Miércoles 10 de Mayo 18:30
  - $I2$: Jueves 08 de Junio 18:30
- 1 Examen:
  - E: Lunes, 03 de Julio 18:30
- Nota de participación:
  - NP: Es distinto para cada sección. Nosotros usaremos un sistema de décimas.

Reglas: Evaluaciones

Abril
10 - 26
Tarea 1
(16 días)

Mayo	
Miércoles 10	15 - 31
Interrogación 1	**Tarea 2**
18:30	**(16 días)**

Junio	
Jueves 8	12 - 30
Interrogación 2	**Tarea 3**
18:30	**(18 días)**

Julio
Lunes 3
Examen
18:30

Reglas: Evaluaciones

¿Cómo apruebo?

La nota final (NF) del curso se calcula en base a la siguiente fórmula:

$$NF = 15\% * I1 + 15\% * I2 + 30\% * E + 10\% * T1 + 10\% * T2 + 10\% * T3 + 10\% * NP$$

Para aprobar el curso, debe cumplir todas las siguientes condiciones:

- Promedio ponderado de evaluaciones mayor o igual a 4.0: $(15\% * I1 + 15\% * I2 + 30\% * E)/60\% \geq 4.0$
- Promedio de tareas mayor o igual a 4.0: $(T1 + T2 + T3)/3 \geq 4.0$
- Nota Final debe ser mayor o igual a 4.0: $NF \geq 4.0$

Si alguna de estas condiciones no es satisfecha, la nota final del curso será el mínimo entre NF y la nota 3.9.

Toda nota (parcial o final) será redondeada a 1 decimal.

Reglas: Inasistencias

En caso de inasistencia se debe justificar ante la Dirección de Pregrado correspondiente, quienes entregarán una carta de inasistencia. Finalmente esta carta debe ser enviada vía e-mail a iic1103@uc.cl

Solo es posible justificar la inasistencia a **una interrogación** y esta nota será reemplazada por la nota del examen.

El plazo para justificar una insasistencia es de 10 días normales (incluyendo días no hábiles) desde ocurrida la inasistencia. En el caso de tener licencia médica, el tiempo comienza a correr desde el reintegro a la universidad.

Reglas: Ética

Código de Honor Código de Honor Código de Honor

Reglas: Ética

- Similaridad de código es:
  - Tomar y usar el código de otra persona del curso.
  - Tener respuestas iguales con otra persona del curso.
  - Trabajar juntos en la tarea.
  - Mirar la solución de otra persona del curso.
  - Discutir en detalle cómo resolver una tarea.
  - Trabajar con un profesor particular en la tarea.
  - Comprar tareas o participar en compra de tareas.

Reglas: Ética

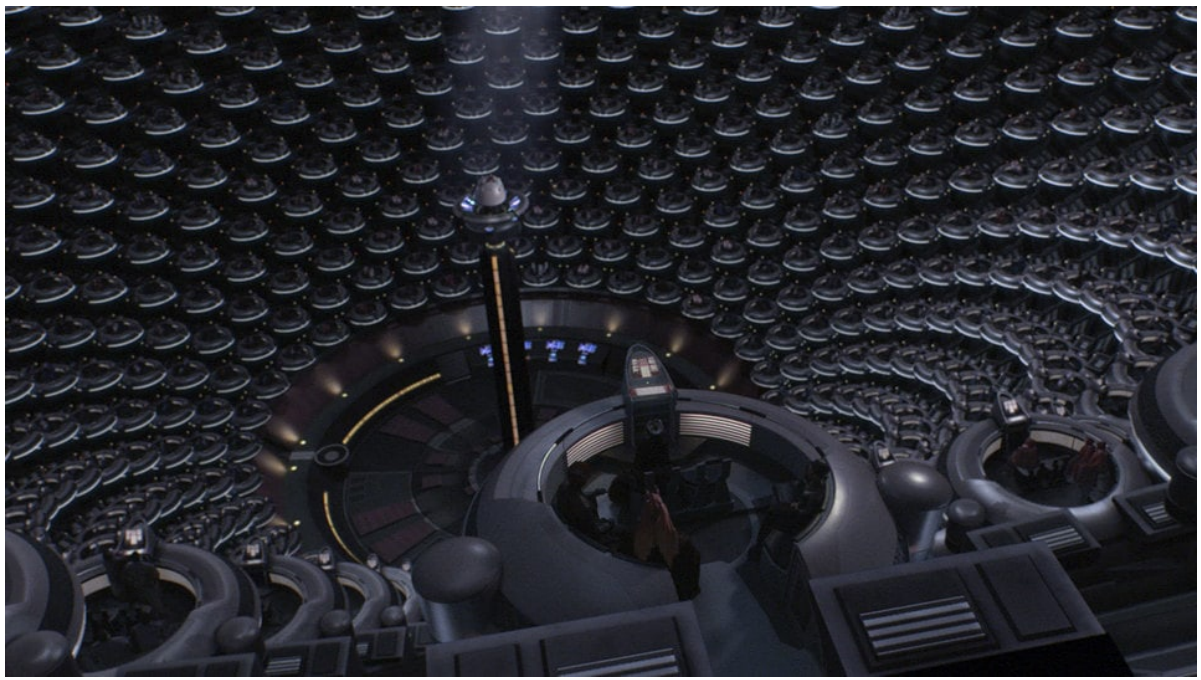
- Similaridad de código es:
  - Tomar y usar el código de otra persona del curso.
  - Tener respuestas iguales con otra persona del curso.
  - Trabajar juntos en la tarea.
  - Mirar la solución de otra persona del curso.
  - Discutir en detalle cómo resolver una tarea.
  - Trabajar con un profesor particular en la tarea.
  - Comprar tareas o participar en compra de tareas.

- No es similaridad o copia:
  - Basarse en código de un repositorio público.
  - Discutir en general cómo resolver un problema.

Reglas: Ética

- Copia es:
  - Tomar y usar el código de otra persona.
  - Tener respuestas iguales con otra persona.
  - Trabajar juntos en la tarea.
  - Discutir en detalle cómo resolver una tarea.
  - Comprar tareas o participar en compra de tareas.

- ¿Qué pasa si copio?



Reglas: Ética

- ¿Qué pasa si copio?
  - Se contacta al estudiante por correo indicando que detectamos similaridad de códigos.
  - Se cita a ambos estudiantes a una sesión privada con TODOS los profesores del curso.
  - Se les da la oportunidad de explicarse.
  - Si existe copia, compra o transacción del trabajo, **se obtiene nota final 1.1 en el curso, SIN OPCIÓN DE RETIRO.**

Ustedes al final del curso:



Ustedes al final del curso:



Bibliografía

- Clases.
- Python v3 Documentation, http://docs.python.org/3/