

SESIÓN 2

Introducción a la Programación en Python

Nivel 2 del Currículo de AOLME

OBJETIVOS

1. Aplicar ideas básicas de Programación en Python.
2. Programar operaciones básicas y variables en Python.
3. Resolver y crear operaciones y expresiones aritmético-algebraicas.
4. Programar un juego para adivinar números usando Python.



Actividades:

- 1 Explorando cómo se programa en Python (print, strings, operaciones elementales).**
- 2. Programar operaciones en Python (ecuaciones lineales, orden de las operaciones y expresiones complejas).**
- 3. Crear tu propio juego de adivinanzas en Python (diseñar, programar y compartir el juego resultante modificando sus códigos base)**

1	2
3	4

Cada actividad incluye 1 o 2 cartas. Un lado en español y el otro con la misma información en inglés. Cada carta tiene cuatro cuadrantes que incluyen una tarea relacionada con el objetivo principal de la actividad. Los números en el recuadro del lado izquierdo describe el orden de las tareas a hacer. La carta debe estar al centro de la mesa, para que los estudiantes tengan acceso y puedan tomar turnos al leerla.

2.1. EXPLORAR LA PROGRAMACIÓN EN PYTHON

Objetivo de la actividad 1:

Aplicar conceptos básicos de la programación en Python.

Materiales para la actividad

1. Hoja de actividades
2. Carpeta: /home/pi/AOLME/Session 2/
3. La libreta “Jupyter” que se accesa por medio de la Terminal.
4. Raspberry Pi y Monitor
5. Cuaderno del estudiante

Interacciones

En esta actividad los estudiantes deben pensar a cerca de cómo programar en una computadora. Es importante que los estudiantes tengan la idea de que programar es como una serie de pasos para lograr algo. Usa ejemplos de la vida diaria y haz que los alumnos hablen y vean como la programación se relaciona en sus vidas. El juego en esta sesión trata de ayudar a los alumnos a ver la programación de esta manera en vez de algo muy complicado. Siempre trata de usar experiencias de los alumnos como un punto de partida y de ahí hacer uso de ideas. Aunque las instrucciones para esta actividad son cuatro, estás sólo describen cómo las ideas pueden ser explicadas. A lo largo de esta actividad hay que proveer un ambiente amigable, de apoyo y participación para todos. Nota quién participa más o menos, presta atención al por qué y trata de tomar acción sobre ello, para que la participación entre todos sea más uniforme. Apoya en todo momento el uso del lenguaje (Español o Inglés) que los estudiantes prefieran utilizar.

Hoja de la Actividad 2.1:

2.1. Explorando cómo se programa en Python



1. Practica programar con Python en jupyter. Abre la terminal > Ve al directorio AOLME y escribe: `jupyter notebook`. Presiona Enter, y en Jupyter abre el folder de la Sesión 2. Selecciona New y luego Python 3.

Copy & Paste

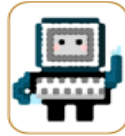
1. Select
2. Copy
3. Paste

Las celdas son espacios donde puedes entrar información. Puedes copiar y borrar celdas

En el documento nuevo, navega Jupyter explorando qué hace cada icono en la barra de tareas en. ¿Qué es una celda?

2. Para jugar al Guessing Game:

- (a) Abre la carpeta o folder de la Sesión 2 en Jupyter
- (b) En 'Task 2', selecciona la celda llamada: EXAMPLE with game code.
- (c) Haz click en 'run'
- (d) Introduce los datos en la parte inferior de la celda.
- (e) Tomen turnos.
- (f) Limpian la celda haciendo click en 'run' para introducir nuevos datos.



Piensa y habla:
¿Cómo funciona?

3. Juega y corre las celdas CELLS A-C en Task 3. ¿Qué hace el comando print?

¿Por qué cambia de color la celda?

Estas cajas son celdas (de Python o texto). Al inicio algunas tienen: `'In []' y otras no. ¿Por qué?`

```

TASK 3: Comparing Python Cells & Using 'print' Commands
# CELL A: When you type "f" at the beginning of a line, it creates a COMMENT.
# The computer does not process comments.
# USE COMMENTS TO DESCRIBE WHAT YOU WRITE.
# Computers process code in order, line by line. See below:
# print(10)
# Write in this cell, click on the play button and run the code. What happens?
# After running this code, color cell and print other numbers.

In [ ]: # CELL B: Strings are bits of programmed text in this cell.
# Try this code and see if, what happens? What at the code below delay?
name = input("What is your name? ")
hi = "hello" + name + "!" # Addition is concatenation
print(hi)

In [*]: # CELL C: Python can process complex expressions like:
# = 100
print(10000 + 10000)
  
```

4. Dialoga & Crea: En base a lo hecho y aprendido en task 3, en Task 4, escriban su propio código. Tomen turnos para escribir.

Sean creativos y disfruten!

¿Qué notas acerca del texto rojo y lo que pasa abajo de cada celda? ¿Qué pasa si cambias el texto rojo?

¿Has jugado al guessing game antes?



AOLME PROJECT - NIVEL 1- SESIÓN 2- 2019

MATERIAL DESARROLLADO POR EL PROYECTO AOLME EN LA UNIVERSIDAD DE NUEVO MEXICO. SE RUEGA NO COPIAR NI DISTRIBUIR, TOTAL O PARCIALMENTE, SIN EXPRESA AUTORIZACION DEL PROYECTO.

Pasos recomendados

Tarea 1: Asegúrate de que para la primera tarea los estudiantes abren el archivo 'Untitled' de la carpeta de la sesión. Anímalos a que escriban y a que prueben todos los íconos de la barra de tareas. Que exploren y dialoguen acerca de sus descubrimientos. Ten en cuenta que podría ser que establezcan relaciones con sus experiencias con otras computadoras. Haz que los estudiantes relaciones el vocabulario cuando hablen de las tareas.

Tarea 2: Cuando jueguen el juego de adivinar el número, deja que los estudiantes tomen turnos para jugar, ¿tal vez en parejas? ¿Tomando turnos al escribir? Usa los colores para hablar de la función de los comandos, textos, strings y comentarios.

Tarea 3: Motívalos para que fijen en que hay distintos tipos de celdas. (celdas de solo texto y de programación). es importante que se den cuenta de la función de "print" y de lo que sucede si no utilizan "print" Déjalos que experimenten. Pon especial atención al texto o "string" rojo puede ser cambiado, inclusive a cualquier lenguaje, pero las palabras (incluso las verdes) no pueden cambiarse. Subraya la idea del commando 'print' y que piensen otras ideas para usar ese comando, aparte de los ejemplos que han hecho. Haz que los alumnos tomen turnos para insertar los datos y ejecutar las celdas. Usa lenguaje corporal si es necesario, especialmente cuando alguien dice algo importante, ya sea algo que ya hayas dicho antes o pídeles que hagan lo mismo, pero ¡hazlo de una forma divertida! Haz que los estudiantes tomen decisiones frecuentemente.

Tarea 4: Anímalos a comentar sobre lo que aprendieron y a anotarlo en el cuaderno.

Contenido: ¿Qué es Python?

Las personas usan lenguajes para comunicarse entre ellas (ej. lenguaje de señas, chino, español, lenguajes indígenas), los programadores también usan lenguajes -o “lenguajes de programación”- para comunicarse con las computadoras. Python es uno de esos lenguajes de programación.

‘Python Shell’ como Calculadora

El ‘Python Shell’ puede usarse como una calculadora para realizar las cuatro operaciones aritméticas básicas como adición (+), sustracción (-), multiplicación (*) o división (/).

Escribe las operaciones sugeridas en la tabla de abajo. Pero, **¿cómo ejecutar las operaciones para obtener la respuesta de la computadora? ... Presiona Enter!**

Variables en Python

¿Haz estudiado álgebra? El álgebra expresa valores y relaciones a través de símbolos usando letras y números. Por ejemplo: $x + 4 = 10$

¿Cuál es el valor de “x”?

- ⊕ Las variables son fundamentales en las matemáticas y la programación.
- ⊕
- ⊕ **En Python, una variable puede ser representada por una letra o una palabra** la que puede contener un valor conocido o desconocido. Por ejemplo:

```
>>> a = 2  Esta variable 'a' tiene el valor de 2
>>> b = a+5  Ahora, la variable 'b' ahora tiene el valor de 7 que equivale al
              valor de 'a', que es 2, más 5.
>>> Juego = "adivinar el número"  Aquí la variable 'Juego' está
                                     representada por una palabra que tiene
                                     el valor: adivinar el número.
```

Distintos tipos de variables

Mira los comandos debajo, ejecútalos usando Python Shell, y escribe las respuestas.

Integ	<pre> INTEGER >>> a = 5 >>> b = int(a) >>> print b </pre>	<pre> INTEGER >>> a = 5.3 >>> b = int(a) >>> print b </pre>
Float	<pre> FLOAT >>> a = 7.1 >>> b = float(a) >>> print b </pre>	<pre> FLOAT >>> a = 6 >>> b = float(a) >>> print b </pre>
String	<pre> STRING >>> a = '10' >>> print a </pre>	<pre> STRING >>> a = 'Hello!' >>> print a </pre>

Principales variables en Python	
Integer	<ul style="list-style-type: none"> ⊕ En programación se le refiere como <code>int</code> ⊕ Se usa para representar valores enteros positivos y negativos. ⊕ La función <code>int()</code> convierte un valor a 'Integer'.
Float	<ul style="list-style-type: none"> ⊕ En programación se le refiere como <code>float</code> ⊕ Se usa para representar valores reales. ⊕ La función <code>float()</code> convierte un valor a un valor real.
String	<ul style="list-style-type: none"> ⊕ Se representa escribiendo los datos entre comillas "" ⊕ Se usa para representar texto, números y símbolos.

2.2. PROGRAMACIÓN DE OPERACIONES NUMÉRICAS CON PYTHON

Objetivos de la actividad 2:

- (a) Programar variables y operaciones básicas en Python.
- (b) Crear y resolver sus propias operaciones y expresiones aritmético-algebraicas.

Materiales para la actividad

1. Hoja de actividades
2. El “Jupyter” notebook, accesible via *Terminal*
3. Carpeta: /home/pi/AOLME/Session 2/
4. Raspberry Pi y Monitor
5. Cuaderno del estudiante

Interacciones:

A lo largo de esta actividad hay que proveer un ambiente amigable, de apoyo y participación para todos. La meta no es ver cuanto saben de matemáticas, sino pensar como usar la programación para las matemáticas. Si es necesario, haz que los equipos hablen acerca de lo que saben de matemáticas y pregúntales de dónde creen que vienen los resultados. Por ejemplo, si alguien inserta un 3 y el resultado es 1.5, explora como eso sucedió en relación con las matemáticas y el código. Nota quién participa más o menos, presta atención al por qué y trata de tomar acción sobre ello, para que la participación entre todos sea más uniforme. Apoya en todo momento el uso del lenguaje que los estudiantes prefieran utilizar.

Hoja de actividad 2.2:

2.2. Programación de Operaciones Numéricas con Python

Nota: The numbers in Card match the Tasks numbers in Jupyter Notebook.

1. Vamos a usar álgebra y matemáticas para programar el *Guessing Game* con Python.



En el *Guessing Game*, no sabemos el número que la persona está pensando. Todo lo que sabemos son las operaciones que les pedimos que hagan. Ahora, veamos cómo las operaciones se relacionan al número que se adivinará.

3. En la CELL 2B en **jupyter**, hay variables (x, y, & z) ¿Qué es una variable? Corre la Cell varias veces para cambia los números y las operaciones, y renombra las variables.

¿Cómo crees que podemos usar las variables y las operaciones para adivinar un número que no sabemos?

2. En la CELL 2A en **jupyter**, corre el código varias veces y discute qué pasa.

```
x = int(input("Enter a number: "))
print("Multiplying x by 2 gives: 2*x =", 2*x)
print("Adding 3 to the result gives 2*x+3 =", 2*x+3)
print(" ")
print("What happened?")
```

Enter a number: x = 3
Multiplying x by 2 gives: 2*x = 6
Adding 3 to the result gives 2*x+3 = 9

Usa otros números y explora cómo se relacionan.

El número que se pone en 'x' es 3. ¿Cómo se relacionan el 6 y el 9 al 3?

4. En la CELL 2C, crea tu propio juego cambiando las operaciones y poniendo números en lugar de '?'. Si lo deseas, re-nombra las variables.



¿Cómo crees que Python nos ayuda a hacer matemáticas?

¿En qué se parecen pensar y programar?

AOLME PROJECT - NIVEL 1- SESIÓN 2- 2019

MATERIAL DESARROLLADO POR EL PROYECTO AOLME EN LA UNIVERSIDAD DE NUEVO MEXICO. SE RUEGA NO COPIAR NI DISTRIBUIR TOTAL O PARCIALMENTE, SIN EXPRESA AUTORIZACIÓN DEL PROYECTO.

Pasos recomendados para esta actividad

Tarea 1: Habla acerca de lo que es el Algebra en matemáticas y haz que los estudiantes establezcan conexiones entre los procesos algebraicos, las variables y la programación. Usa el Código en Jupyter para pensar acerca de estas ideas. Usa el Código como evidencia para apoyar estos argumentos/ideas.

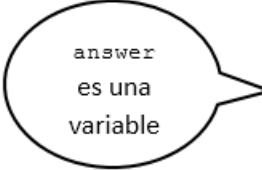
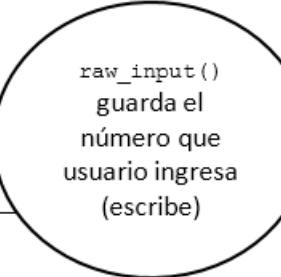
Tarea 2: Motiva a los estudiantes a usar vocabulario relacionado al referirse al texto rojo o verde, como strings, comentarios o comandos mientras operan con las variables. De igual manera, asegúrate que el concepto de Variable sea entendido por todos. Usa predicciones como una manera de que los estudiantes piensen acerca de las matemáticas y programación. Asegúrate que todos los estudiantes participen en este proceso.

Tarea 3: Asegúrate de que los estudiantes entienden el hecho de que podemos determinar variables de muchas maneras distintas; también de que identifiquen la manera utilizada en cada caso y de que hablen sobre ello. Insiste en que, cada vez que vayan a ejecutar una celda, hagan primero una predicción sobre qué hará cada línea del código, y en que, posteriormente, comprueben lo que realmente hace. Cambiar algunos de los números del código podría ser una buena idea para ayudarles a entender mejor el funcionamiento de los códigos.

Tarea 4: A la medida que vayas creando el juego, asegúrate de incluir las ideas de todos. Revisen entre todos que las matemáticas y la programación sean correctas. Si algo sale mal, deja que los alumnos resuelvan el problema.

Interviene solo cuando es necesario o cuando los alumnos no puedan descifrar el problema.

Contenido: Flujo de datos en un Sistema de computadora.

Función	Significado
print	print muestra la expresión que ingresas. Solo escribe el texto en comillas después de la palabra <code>print</code> . Por ejemplo, <code>print "Piensa en un número entre el 1 y 10"</code>
raw input() 	raw_input es una función que guardará el texto y lo almacenará en una variable <code>string</code> . Por ejemplo, <code>print "¿Cuál es el resultado?"</code> <code>respuesta = raw_input()</code> 

Ahora veamos a la función 'integer' en la siguiente table.

Función	Significado
int()	La función int() convierte una variable (en el ejemplo anterior es la palabra "respuesta") en un entero (número positivo o negativo). Inicialmente, la variable <code>respuesta</code> es un <code>string</code> (texto o simbolo) cuando es almacenado por la función <code>raw_input()</code> . Sin embargo, necesitamos convertir la variable <code>respuesta</code> en un número, para que pueda ser dividida por otro número, en este caso, el número 3. <code>print "¿Cuál es el resultado?"</code> <code>respuesta = raw_input()</code> <code>print "Tu número inicial fue:"</code> <code>print int(respuesta)/3</code>

¡Aprendimos sobre las funciones, variables, y operaciones matemáticas básicas y eso es suficiente para comenzar a programar el juego matemático en Python!

2.3. CREA TU PROPIO 'NUMBER GUESSING GAME' EN PYTHON

Objetivos de la actividad 3:

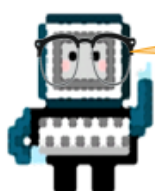
Programar 'number guessing game' usando Python.

Materiales para la actividad:

1. Hoja de actividad
2. Carpeta: /home/pi/AOLME/Session2
3. El "Jupyter" notebook que se accesa via Terminal
4. Raspberry Pi kit
5. Cuaderno de estudiante

Hoja de actividad 3:

2.3. Crea tu propio 'Number Guessing Game' en Python



¿Qué hace que este juego funcione?

1. El Juego: **Number Guessing Game.** Para que el juego funcione, debemos pensar en operaciones inversas. ¿Por qué necesitamos operaciones inversas para que el number guessing game funcione?

2. **Operaciones inversas:** la multiplicación es inversa a la división. Considera esto para el juego:

Dame un número x
Multiplícalo por 2. $2*x$
Divídelo por 2. $(2*x)/2=x$

Luego de operar, el resultado vuelve a ser x . Dividimos por 2 porque es inverso a multiplicar por 2 y así podemos deshacernos de lo que multiplicamos por 2. ¿Por qué queremos deshacernos de valores que entramos?

¿Cuál es la operación inversa a la suma? ¿Por qué?

3. **Reflexiona:** ¿Qué va a pasar en este juego?

Dame un número x
Súmale 3 $x+3$
Réstale 3. $(x+3)-3=x$



¿Qué valores o números combinarías en el guessing game?

4. **Diseña y programa** en equipo un *number guessing game*. Escriban los códigos turnándose el teclado. Guarden el trabajo. Usen el código en la celda 2.3 para iniciar su juego. Al terminar, pidan a alguien de otro equipo que pruebe el juego que crearon.

Prueben si el juego funciona usando papel.



Discutan y escriban en su cuaderno:

- ¿Qué aprendiste acerca de la programación de variables y ecuaciones?
- ¿Cómo las variables y las operaciones hacen que el juego, guessing game, funcione?.

AOLME PROJECT - NIVEL 1- SESIÓN 2- 2019

MATERIAL DESARROLLADO POR EL PROYECTO AOLME EN LA UNIVERSIDAD DE NUEVO MEXICO, DE SIN PREVIA AUTORIZACIÓN DEL PROYECTO.

Pasos Recomendados para esta actividad

Tarea 1: Anima a los estudiantes a reflexionar sobre cómo los números con las operaciones inversas eliminan números. Escribe las ecuaciones mientras lo hacen. Pregúntales que conexiones ven entre las operaciones y las operaciones inversas.

Tarea 2: Pídeles que hagan pruebas en papel, eliminando números hasta "adivinar" el número. Compara con ellos cómo la terminal es parecida o diferente del GUI.

Tarea 3: Promueve la colaboración al escuchar y ayudarse mutuamente. Intenta no darles las soluciones, más bien hazles preguntas que estimulen su razonamiento. Invita miembros de otros equipos para jugar el juego de tu equipo.

Tarea 4: Motívalos para que colaboren y la pasen bien, desarrollando su creatividad y comunicación. Al final, pedir a los estudiantes que describan por lo menos 3 ideas que aprendieron en la sesión. Motívalos a decir algo más aparte de que ¡fue divertido!