

Estructura de Datos y Programación Orientada a Objetos

Módulos, Paquetes

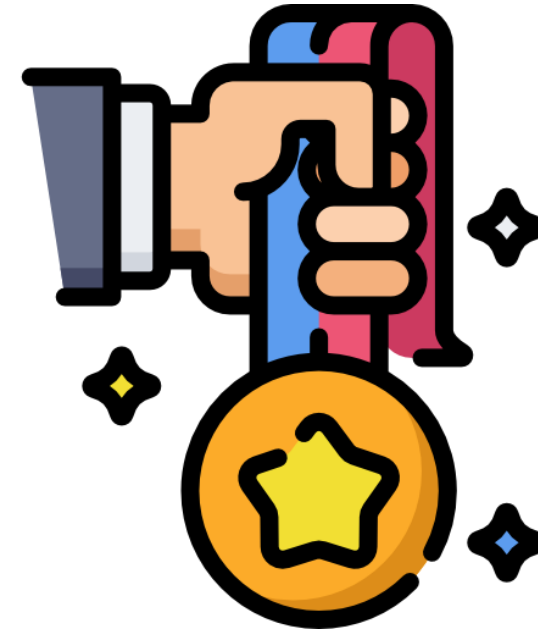
Semana 1



Logro de aprendizaje



- Utiliza módulos existentes, crea módulos nuevos, los agrupa en paquete a través de un lenguaje de programación.



Contenidos



MÓDULOS,
PAQUETES



El uso de módulos



Algunos módulos útiles

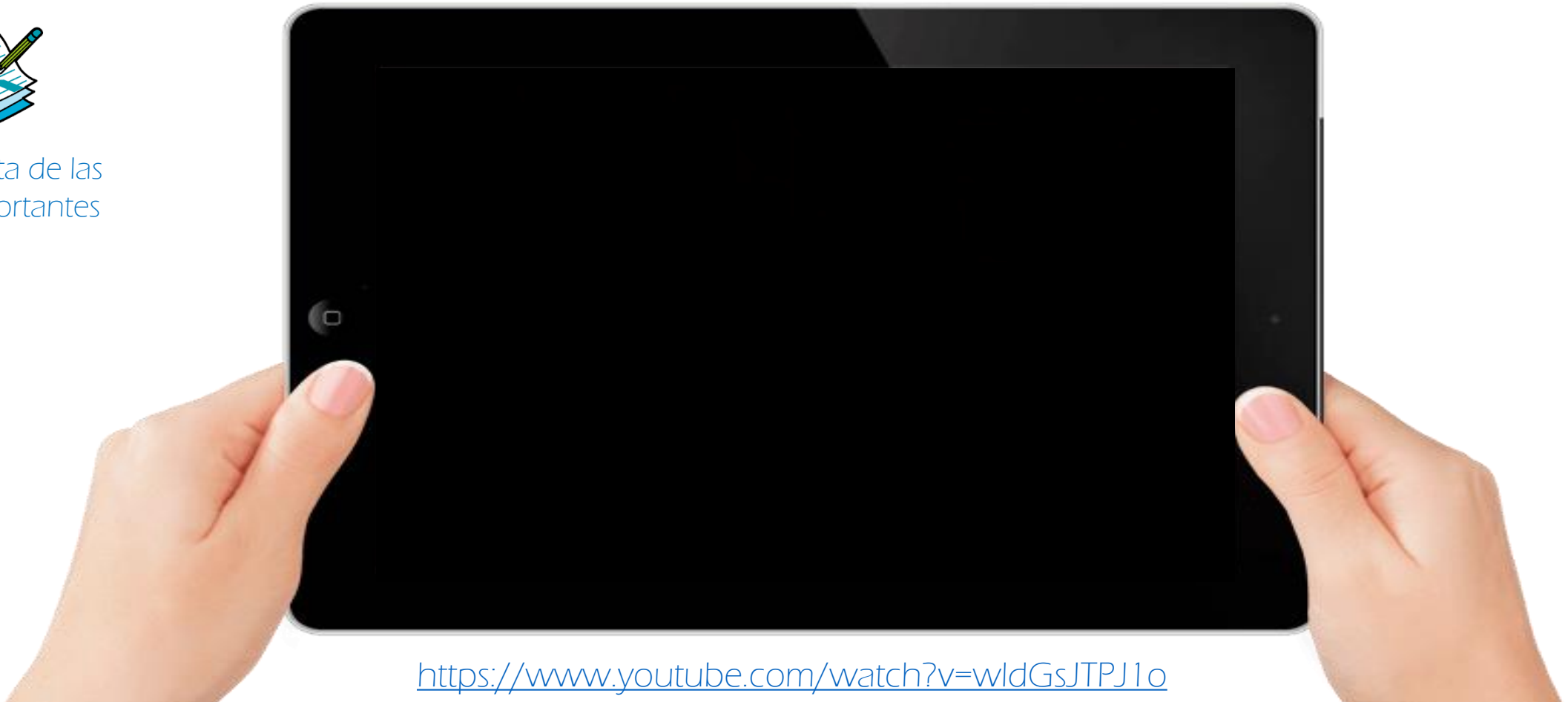


¿Qué es un paquete?

Observemos el siguiente vídeo



Tomar nota de las
ideas importantes



<https://www.youtube.com/watch?v=wldGsJTPJ1o>

INICIO

Porqué utilizar Python



5 RAZONES PARA APRENDER PYTHON



1. RECOMENDADO COMO PRIMER LENGUAJE DE PROGRAMACIÓN



Gracias a su sintaxis sencilla y limpia, **es un lenguaje rápido de aprender.**

2. MULTIPARADIGMA



Se puede **programar orientando a objetos** o utilizando programación **funcional o imperativa.**

3. DESARROLLO RÁPIDO



Puedes crear aplicaciones **en menos líneas de código** que muchos otros lenguajes.

5. SEGURIDAD INFORMÁTICA



Es uno de los lenguajes **más utilizados** y **adoptado** por la **industria de la seguridad informática**

4. MACHINE LEARNING



Python es el lenguaje más usado en **modelos de Machine Learning.**

CAMPOS DE APLICACIÓN DE PYTHON

Python es el lenguaje más usado, es fácil de aprender y tiene muchos campos de aplicación. **¿Qué esperas para aprenderlo?**



SEGURIDAD INFORMÁTICA



Programa scripts que ejecuten pruebas automáticas para detectar vulnerabilidades.

DESARROLLO WEB



Crea apps web con frameworks como Django, Flask, Pyramid, etc.

TESTING Y QA



Automatiza tests de código y de funcionalidades.

BIG DATA Y DATA SCIENCE



Extrae, procesa, almacena (ETL) y analiza grandes cantidades de datos.

VIDEOJUEGOS



Crea videojuegos con los frameworks: PyGame, PyOpenGL, Blender, etc.

MACHINE LEARNING



Escribes modelos de machine learning con librerías como SciKit, SciPy, etc.

Porqué utilizar Python



SINTAXIS BÁSICA DE PYTHON

Python se usa en machine learning, web, seguridad informática, data science, cloud computing y tiene una de las sintaxis más sencillas.



```
1 #!/usr/bin/python3
```

Comentarios con # al inicio

```
2 import sys
```

Importación de módulos

```
3 def holamundo():
```

Las funciones se definen con **def** y el nombre de la función.

```
4     for i in range(1, 10):
```

Los ciclos **for** pueden indicar un rango o una estructura de datos para recorrer.

```
5         nombre = 'EDteam'
```

```
6         print("Hola Mundo " + nombre)
```

```
7 if __name__ == "__main__":
```

No se necesita una palabra reservada para **declarar una variable** ni indicar el **tipo de dato**.

```
8     holamundo()
```

Condicional

Ejecución de función

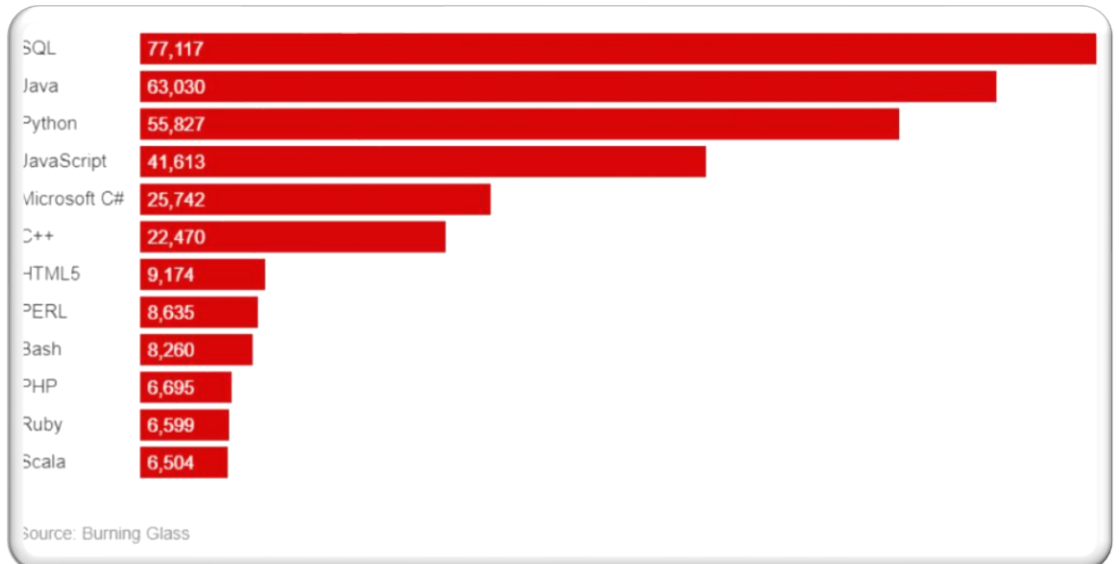
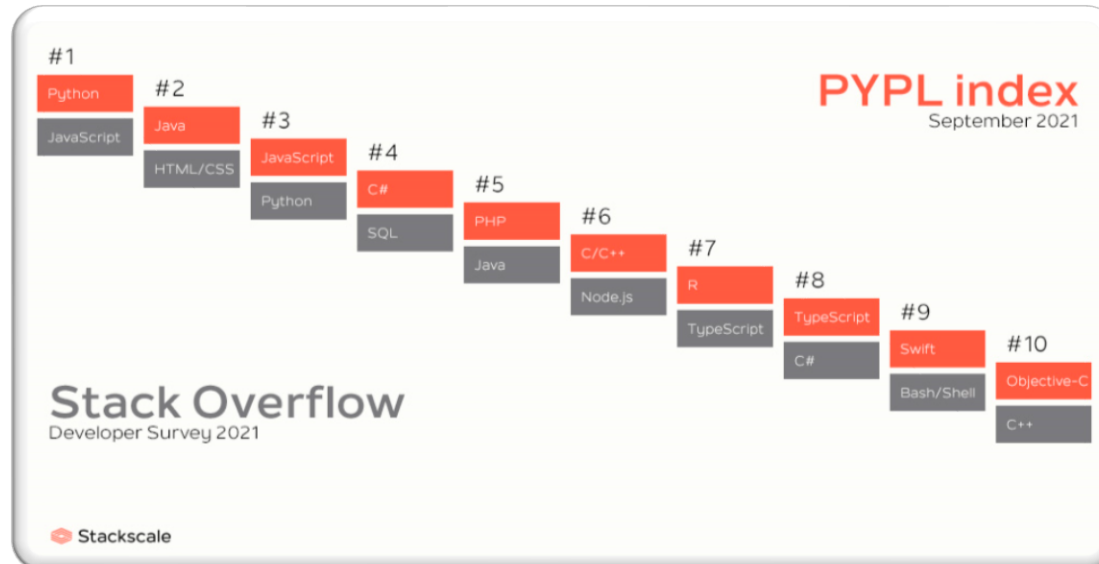
Imprimir salida en consola



Python no usa punto y coma o llaves. Usa saltos de línea e indentación

En python cualquier función o sentencia (**def, for, if, while**) termina con :

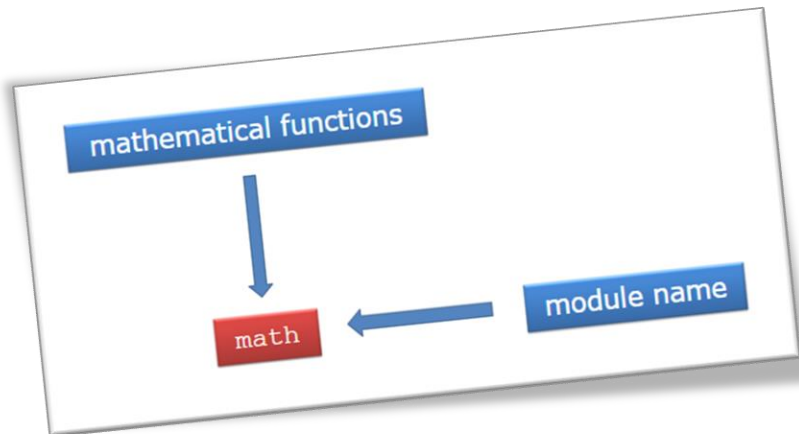
Porqué utilizar Python



¿Qué es un módulo?



- Los módulos son archivos con extensión `.py`, que poseen su propio espacio de nombres y que pueden contener variables, funciones, clases e incluso otros módulos.
- Sirven para organizar y reutilizar el código (`modularización` y `reutilización`).



Importación de un módulo



PRIMER MÉTODO:

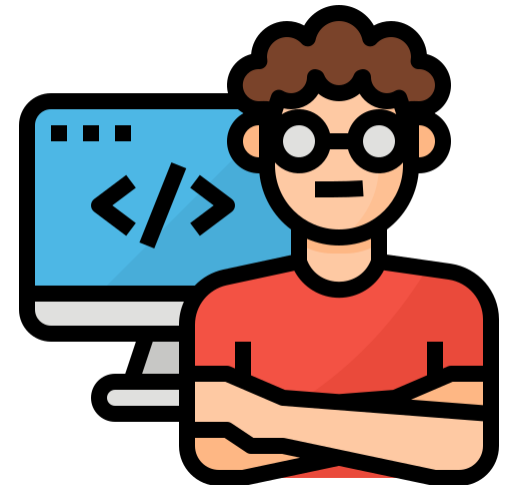
```
import <nombre módulo>
```

```
import math
```

Ejemplo

```
import modulo  
modulo.funcion(argumentos)
```

```
import math  
math.sin(0)
```



Importación de un módulo



SEGUNDO MÉTODO:

```
from <nombre módulo> import  
<elemento1>, <elemento2>
```

```
from math import pow, sqrt
```

Ejemplo

```
from modulo import funcion  
funcion(argumentos)
```

```
from random import uniform  
uniform(0,1)
```



Importación de un módulo



TERCER MÉTODO:

```
from <nombre módulo> import  
<elemento1> as <alias>
```

```
from math import e as euler
```

Ejemplo

```
from modulo import función  
as mifun  
  
mifun(argumentos)
```

```
from random import randint  
as rnd  
  
rnd(5, 10)
```



Importación de un módulo



CUARTO MÉTODO:

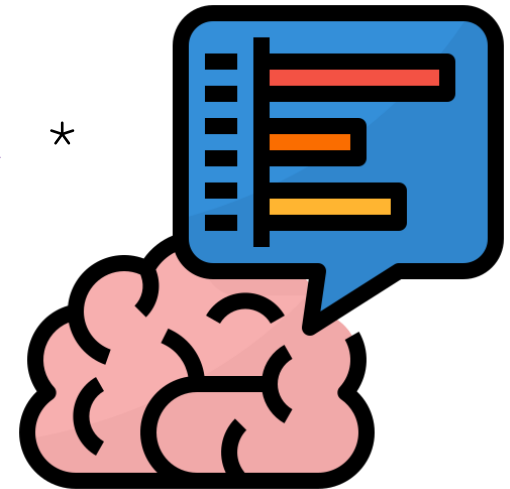
```
from <nombre módulo> import *
```

```
from math import *
```

Ejemplo

```
from modulo import *  
funcion(argumentos)
```

```
from random import *  
uniform(0,1)
```



¿Qué es un paquete?



- Los **paquetes** son ficheros que contienen definiciones que se pueden importar en otros **scripts** para reutilizar sus funcionalidades.
- Utilizar **paquetes** nos ofrece varias ventajas. En primer lugar, nos permite unificar distintos módulos bajo un mismo nombre de paquete, pudiendo crear jerarquías de módulos y submódulos, o también subpaquetes.





ACTIVIDAD 1:



Consigna: Trabajo Individual: Instalación de Visual Studio Code

- Cada alumno tendrá que instalar el Programa Visual Studio Code en sus respectivos ordenadores.



Recursos: Pc o Laptop, Instalador de Visual Studio Code

<https://code.visualstudio.com/>



Tiempo: 15 minutos

Aplicación:



ACTIVIDAD 2:



Consigna: Trabajo Individual: Resolución de Ejercicios

- Cada alumno tendrá que resolver en Python los ejercicios propuestos por el docente:



Recursos: Pc o Laptop, Instalador de Python

<https://www.python.org/>



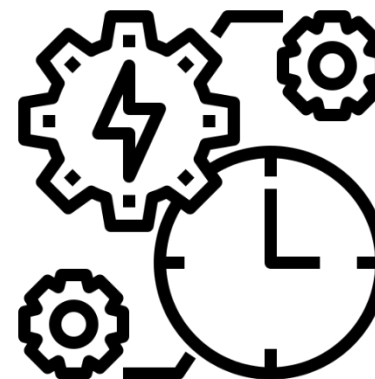
Tiempo: 15 minutos

Ejercicios



EJERCICIO 1

- Realizar un programa que simule tirar dos dados y luego muestre los valores que aparecieron. Si la suma de dichos números es igual a 9 mostrar un mensaje de “Has ganado” sino mostrar “Has perdido”.



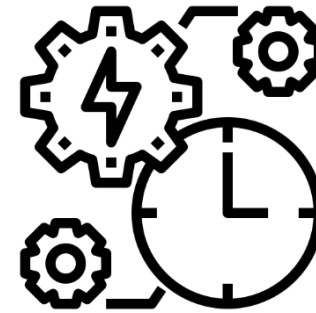
```
El primer dado vale: 1  
El segundo dado vale: 1  
La suma de los dados es: 2  
Has perdido
```

Ejercicios

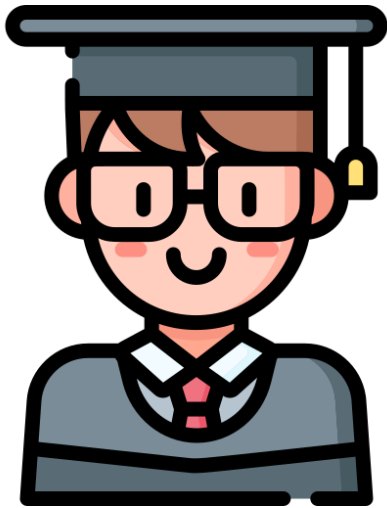


EJERCICIO 2

- Desarrollar un programa que calcule el área de las siguientes figuras: cuadrado, rectángulo, trapecio. Dividir el programa en 4 partes, 3 módulos, cada módulo tendrá una función que calcule el área de un cuadrado, de un rectángulo, de un trapecio y por otro lado un archivo que permita la ejecución de dichas funciones.



```
Ingrese el lado del cuadrado: 4
El área del cuadrado es:  16
=====
Ingrese la base del rectángulo: 3
Ingrese la altura del rectángulo: 4
El área del rectángulo es:  12
=====
Ingrese la base mayor del trapecio: 2
Ingrese la base menor del trapecio: 4
Ingrese la altura del trapecio: 5
El área del trapecio es:  15.0
=====
```



1

¿Qué aprendimos hoy?

2

¿Por qué el tema tratado es importante en mi formación como programador?



**Muchas Gracias
por su atención**

¿Alguna pregunta?

¿No?

Excelente