

# VJ1203 – Programación I

## Práctica 2: Bucles\*

### Curso 2014/15

#### Ejercicios con números y bucles

**Ej. 1 — CosaLosaWosa.** Escribe un programa que utilice un bucle y condicionales para visualizar los números del 1 al 110 en 10 líneas (o sea, 11 números por línea), de la siguiente manera: si el número es múltiplo de 3 deberá escribir ‘‘Cosa’’, si es múltiplo de 5 deberá escribir ‘‘Losa’’, si es múltiplo de 7 deberá escribir ‘‘Wosa’’, para múltiplos de 3 y 5 escribirá ‘‘CosaLosa’’, etc.

El resultado de la ejecución del programa sería el siguiente:

```
1 2 Cosa 4 Losa Cosa Wosa 8 Cosa Losa 11
Cosa 13 Wosa CosaLosa 16 17 Cosa 19 Losa CosaWosa 22
...
```

**Ej. 2 — Calcular la raíz cuadrada entera de un número.** Escribe un programa que utilice un bucle para calcular la raíz cuadrada entera de un número entero positivo sin utilizar la función `sqrt`. La raíz cuadrada entera de un número se define como el mayor entero cuyo cuadrado es menor o igual que el número. Por ejemplo, para calcular la raíz cuadrada entera de 27 debemos probar distintos números empezando por el 1 y quedarnos con el 5, porque  $5^2 \leq 27$  pero  $6^2 > 27$ .

#### Ejercicios con el módulo *turtle* y bucles *for*

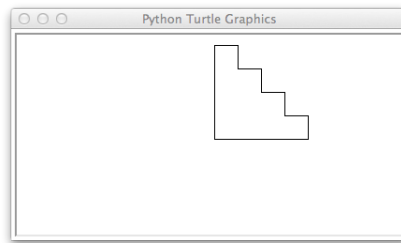
**Ej. 3 — Visualizar distintos tipos de polígonos regulares.** Escribe un programa que utilice el módulo *turtle* y un bucle *for* para dibujar en la ventana gráfica distintos tipos de polígonos regulares, concretamente: triángulo, cuadrado, pentágono y hexágono. El número de lados (o sea, el tipo de figura) y la longitud del lado se leerán de teclado en cada ejecución del programa.

Pista: Recuerda que en los polígonos regulares todos los lados miden lo mismo y todos los ángulos son iguales.

**Ej. 4 — Visualizar los peldaños de una escalera.** Escribe un programa que utilice el módulo *turtle* y un bucle *for* para dibujar en la ventana gráfica una escalera, por ejemplo como se muestra en la figura a continuación. El número de peldaños y la longitud de los mismos se leerán de teclado en cada ejecución del programa.

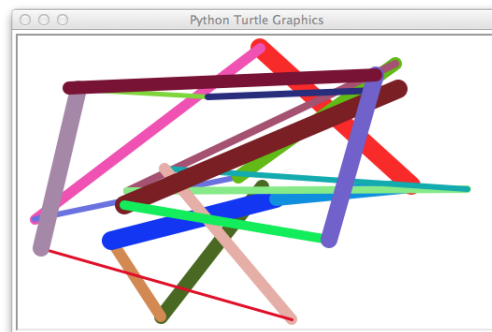
---

\*La práctica 2 tiene una duración de dos sesiones.



**Ej. 5 — Visualizar una serie de puntos y un círculo centrado en el origen.** Escribe un programa que pida al usuario un valor flotante  $r$ , correspondiente al radio de un círculo, y un valor entero  $n$ . Usando el módulo *turtle* y un bucle *for* el programa deberá dibujar en la ventana gráfica el círculo centrado en el origen de coordenadas  $(0, 0)$  y a continuación leer de teclado las coordenadas de una serie de  $n$  puntos en el plano XY, dibujando un punto cada vez que lee unas coordenadas. Además, al terminar el programa deberá imprimir un mensaje con el número de puntos que se han dibujado dentro del círculo.

**Ej. 6 — Visualizar “arte aleatorio”.** Escribe un programa que utilice el módulo *turtle* y un bucle *for* para dibujar en la ventana gráfica una serie de trazos multicolor aleatorios. El número de trazos se leerá de teclado en cada ejecución del programa, dando lugar a sendas iteraciones en las que el programa deberá: (1) generar un grosor aleatorio (entre 1 y 20) para el lápiz; (2) generar un color aleatorio a partir de unos valores (entre 0 y 1) para el rojo, verde y azul; (3) generar unas coordenadas en el plano XY aleatorias (dentro de la pantalla); y (4) utilizar los métodos del módulo *turtle* para cambiar el color y el grosor del lápiz e ir a la posición indicada por las coordenadas dibujando.

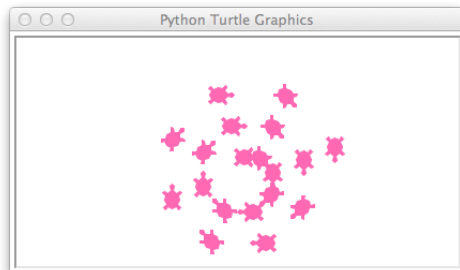


Pista: Para generar un entero aleatorio para el grosor puedes usar la función *randint*, por ejemplo así: `grosor = randint(1, 20)`. Puedes usar también *randint* para generar unas coordenadas aleatorias. Y para generar un número aleatorio entre 0 y 1 para los colores puedes usar la función *random*, por ejemplo así: `redvar = random()`. Estas funciones están incluidas en el módulo *random*.

## Ejercicios con el módulo *turtle* y bucles *while*

**Ej. 7 — Visualizar una espiral de tortugas.** Escribe un programa que utilice el módulo *turtle* y un bucle *while* para dibujar una espiral de tortugas que se ajuste en la medida de lo posible al tamaño de la ventana gráfica, por ejemplo como se

muestra en la figura a continuación. La distancia inicial entre las tortugas y su ángulo de giro se leerán de teclado en cada ejecución del programa.



Pista: Para las marcas de tortuga puedes usar los métodos `shape` y `stamp` del módulo `turtle`.

**Ej. 8 — Visualizar la trayectoria de un “pirata borracho” encerrado en un perímetro de seguridad.** Un pirata borracho al caminar hace un giro aleatorio, avanza un paso de longitud 20, vuelve a hacer un giro aleatorio, vuelve a avanzar un paso de 20, y así sucesivamente. Escribe un programa que utilice el módulo `turtle` y un bucle `while` para dibujar la trayectoria del pirata borracho mientras esté dentro (o hasta que se salga fuera) del perímetro de seguridad circular que se haya definido para él, y mientras no haya superado un número máximo de pasos dado. El radio del círculo y el número máximo de pasos se leerán de teclado en cada ejecución del programa. El programa deberá imprimir un mensaje de alerta si el pirata intenta salir del círculo.

## Ejercicios con el módulo `turtle` y bucles

**Ej. 9 — Visualizar varias pasadas de “arte aleatorio”.** Partiendo del programa del **Ej. 6**, escribe un programa que utilice el módulo `turtle` y bucles `for` y `while` para dibujar en la ventana gráfica varias pasadas de trazos multicolor aleatorios. El número de trazos a dibujar en cada pasada (que será el mismo para todas) se leerá de teclado en cada ejecución del programa. Después de dibujar cada pasada el programa deberá preguntar al usuario si desea continuar dibujando otra, y si no deberá terminar.

**Ej. 10 — Visualizar distintos tipos de polígonos regulares con círculos los bordes.** Partiendo del programa del **Ej. 3**, escribe un programa que utilice el módulo `turtle` y bucles `for` y `while` para dibujar en la ventana gráfica distintos tipos de polígonos regulares adornados con círculos los bordes, por ejemplo como se muestra en la figura a continuación. El número de lados (o sea, el tipo de figura), la longitud del lado y el radio de los círculos (el mismo para todos) se leerán de teclado en cada ejecución del programa.

(continúa en la página siguiente)

