Día 2

Temas:

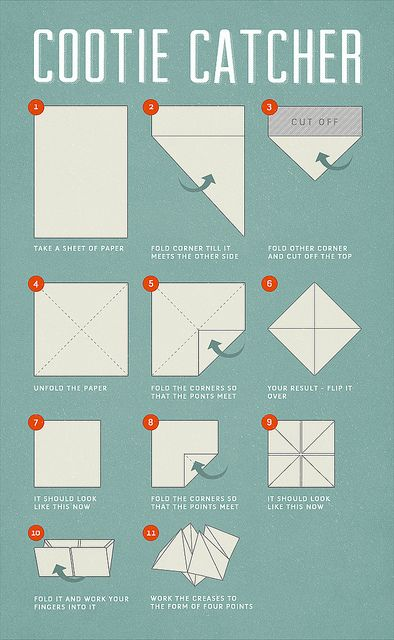
1. Repaso if
   1. else
   2. elif
2. Input
3. Loops
   1. While
      1. Random
      2. Break
      3. Continue
   2. For (Introduction)

Actividades de la Mañana [9:00-12:00]

1. Sillas musicales

Ejercicio relacionado con temas de programación *If* y *else*.

* 1. Materiales: sillas (para cada alumna), una bocina.
  2. Ejercicio grupal.
  3. Duración: 15-20 mins
  4. Instrucciones:
     1. Formar una fila de sillas volteadas de manera intercalada, debe haber una silla por alumna al inicio del juego.
     2. Poner música.
     3. Mientras que la música suene, deberán dar vueltas a las sillas bailando.
     4. Al apagar la música deberán encontrar una silla desocupada y sentarse.
     5. al final de la ronda se elimina una silla y se inicia de nuevo.
     6. La participante que se quede sin asiento será eliminada.

1. Piojitos
   1. Materiales: una hoja de papel para cada niña, lápices de colores y plumones.
   2. Actividad Individual
   3. Duración: 5 -10 mins
   4. Instrucciones:
      1. 
      2. Decorar el piojito
2. Loteria

Tema relacionado con temas de programación *if* y *else*.

* 1. Materiales: juego de Lotería.
  2. Ejercicio grupal
  3. Duración: 25-30 mins.
  4. Instrucciones:
     1. Dar un tarjetón de lotería a cada niña
     2. mostrar las cartas y nombrarlas en orden de aparición.
     3. las niñas deberán marcar las cartas extraídas en sus tarjetones con fichitas

1. Diagrama de flujo actuado
   1. Materiales:
   2. Actividad grupal.
   3. Duración: 15-20 mins
   4. Instrucciones:
      1. Crear una situación hipotética en un restaurante.
      2. Formar equipos dependiendo del número de niñas.
      3. Entre ellas deberán de pensar en todas las opciones posibles en la situación para después actuar según las respuestas de la instructora.
2. Canasta:

Tema relacionado con temas de programación *if* y *else*.

* 1. Duración: 15 minutos
  2. Materiales:
     1. Pelotas pequeñas
     2. Una mesa
     3. Botes medianos
  3. Instrucciones:
     1. Las chicas se separan en dos grupos y hacen filas
     2. Los grupos se colocan al extremo de cada mesa
     3. El objetivo es meter las pelotas dentro de los botes del equipo contrario
     4. Si la chica no logra meter una pelota en el bote de su adversario se deberá ir al final de la fila a esperar jugar de nuevo. En caso de que se logre insertar la pelota dentro del bote de su contrincante éste deberá ser retirado, al final el equipo que se quede sin ningún bote primero será el perdedor.

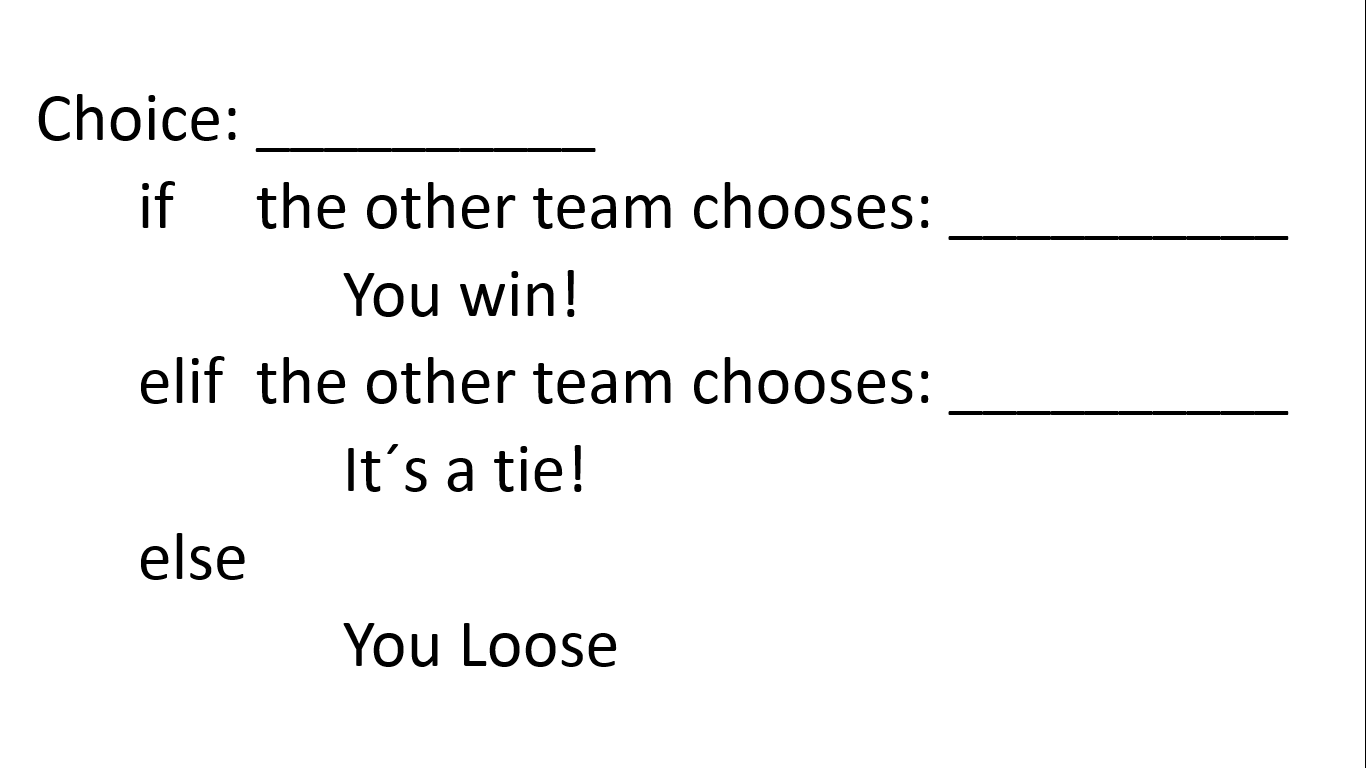
1. El barco se hunde

* El Facilitador indica al grupo de chicas que tienen que formar equipos compuestos por un número exacto de personas que pueden entrar en cada “barco”. Si tienen más o menos personas, se declara hundido el barco.
* El Facilitador cambia el número de personas y así se prosigue hasta que lo considere conveniente.
* Deben darse unos cinco segundos para que se formen los barcos, antes de declararlos hundidos. El Facilitador debe dar las órdenes rápidamente para hacer la dinámica ágil y sorpresiva.

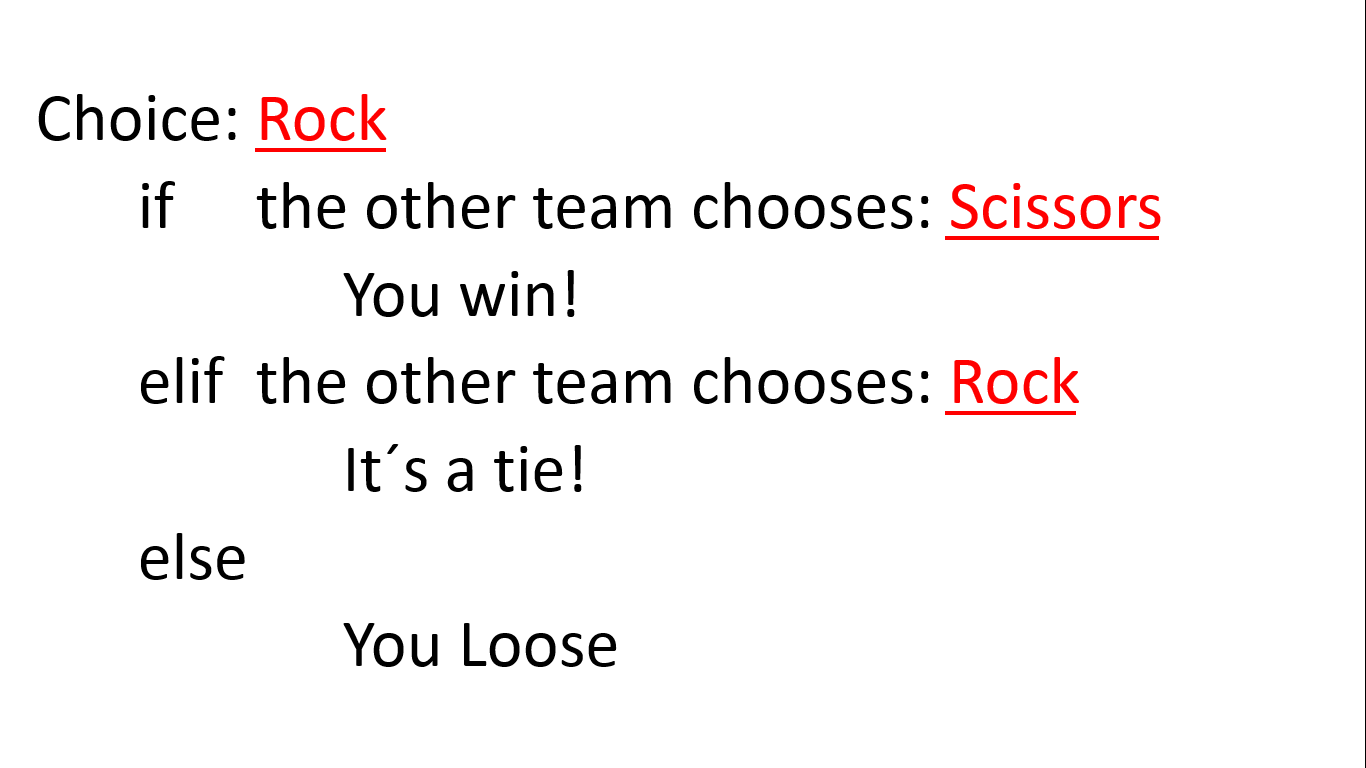
1. Piedra Papel o Tijera

Ejercicio Relacionado con temas de programación *Else* y *Elif*.

* 1. Materiales: Pizarrón, plumones rojos y negros.
  2. Ejercicio en Parejas
  3. Duración: 5 -15 minutos.
  4. Instrucciones:
     1. Cada persona debe formar en el pizarrón el siguiente esqueleto



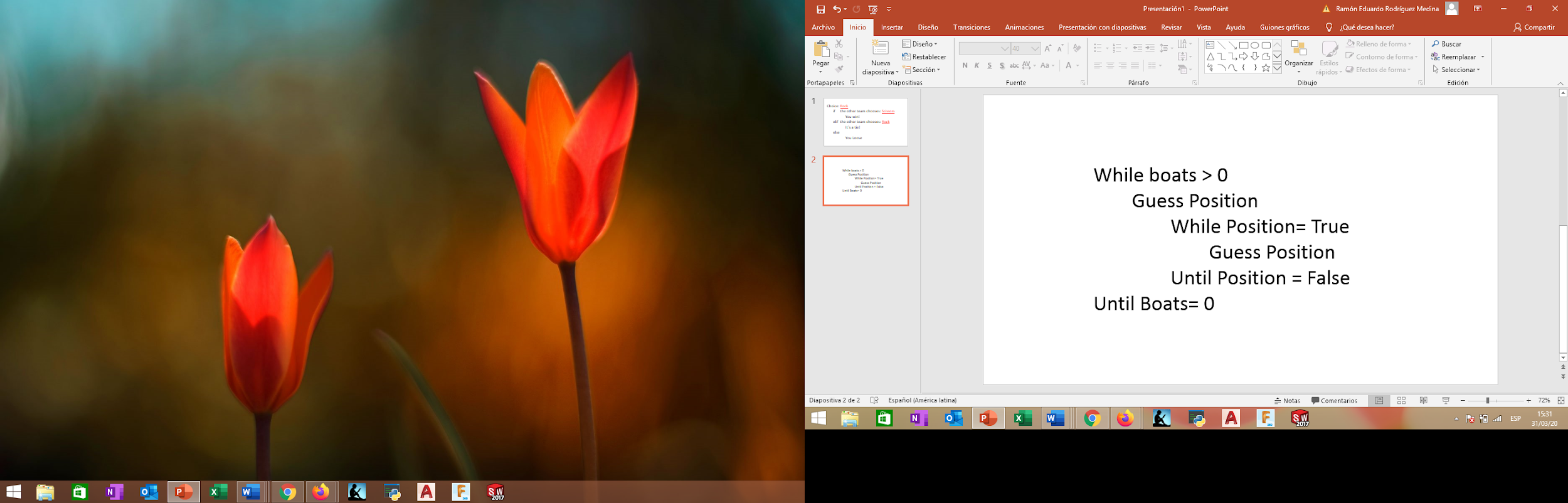
* + 1. todas deberán jugar una ronda de piedra, papel o tijera con sus respectivas parejas, y conforme se de el juego rellenar el esqueleto. Ejemplo:



1. Batalla Naval:

Actividad relacionada con los temas de programación *Nested Loops* y *While loops*. (Podría incluirse el tema *Random*)

* 1. Materiales:
     1. Dos Plantillas Cuadriculadas (representarán las casillas)
     2. Un “muro separador”
     3. Dos plumones.
     4. Barcos:
        1. 3 barcos de 2 casillas
        2. 2 barcos de 3 casillas
        3. 2 barco de 4 casillas
        4. 1 barco de 5 casillas
     5. Apoyo visual:

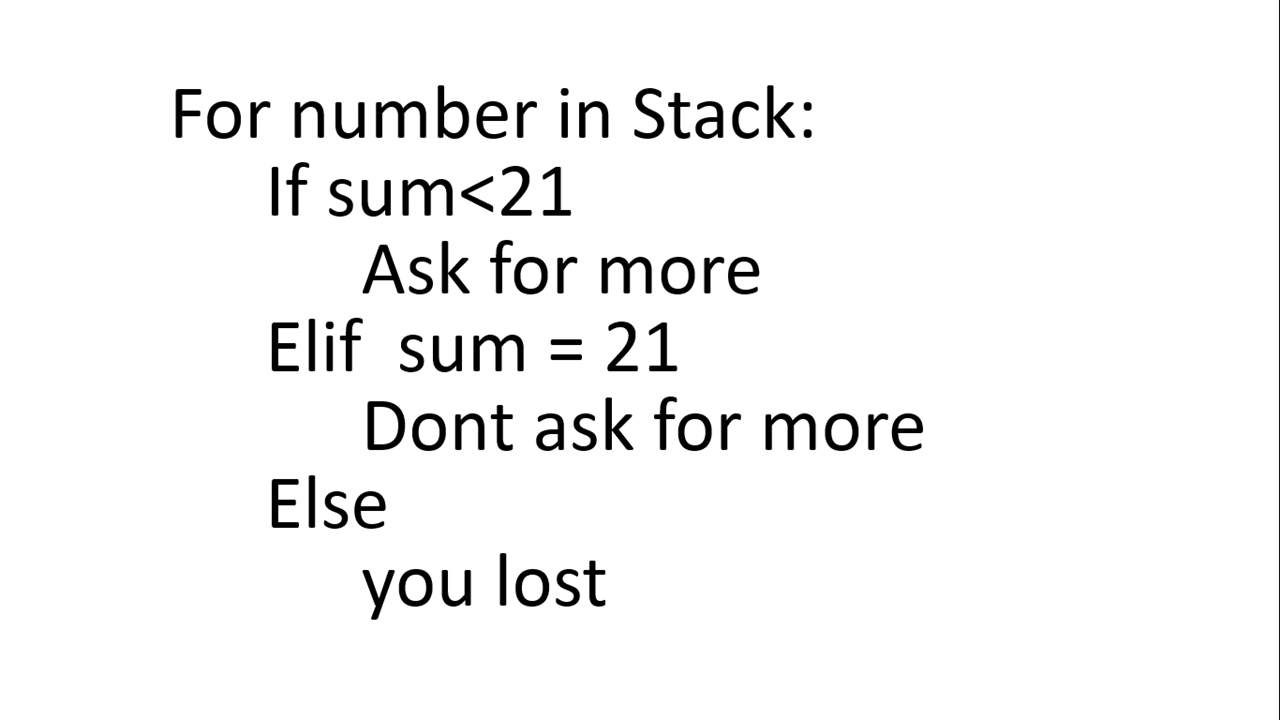


* 1. Dividir al grupo en dos equipos.
  2. Duración: 30-40 minutos si las plantillas son muy grandes.
  3. Instrucciones:
     1. Cada equipo coloca sus barcos en las casillas del tablero. El tablero del equipo contrario no puede ser visto.
     2. El primer equipo debe enunciar las coordenadas que desea atacar (x,y).
        1. Si en las coordenadas se encuentra un barco o parte de él, entonces el barco recibe el daño y el jugador da una nueva coordenada.
        2. Si en las coordenadas no se encuentra un barco, entonces el equipo contrario dice "AGUA" para hacer saber al equipo en turno que ha fallado, terminando su turno.
     3. Cuando un barco ha recibido un daño en cada casilla que ocupa, entonces el barco se habrá hundido. El equipo del barco dañado debe decir "BARCO HUNDIDO".
     4. Todas las coordenadas se marcan en el tablero para que no se repitan. Cuando una coordenada “golpea” un barco del equipo contrario se coloca una equis. Cuando la coordenada no “cae” sobre un barco, se marca con un punto.
     5. Gana el equipo que hunda primero los 8 barcos de su enemigo.

1. Blackjack

Actividad relacionada con temas de programación *for loops* e *input*.

* 1. Materiales: baraja de cartas, apoyo visual:



* 1. Actividad grupal
  2. Duración: 25 - 30 mins
  3. Instrucciones:
     1. Este juego consiste en enfrentarse de forma individual a la casa, comparando su mano con la de cada jugador, intentando conseguir 21 puntos o el número más cercano posible sin pasarse. Para conseguir dicha puntuación se suman los valores de dos cartas que se reparten de inicio a cada jugador con los de aquellas nuevas cartas que, opcionalmente, se podrán añadir en el turno de juego.
     2. Se hace una primera ronda, en la que se entregan las dos primeras cartas aleatorias a cada jugador.
     3. Acto seguido se hace una segunda ronda, en la que el jugador puede pedir otra carta o no; se prosigue con las rondas hasta que ningún jugador pida cartas.
     4. Después se descubren las cartas y se comparan con las de la casa. Si la casa suma 21, gana, incluso contra los jugadores que también tengan 21, si la casa suma menos de 21, pierde contra todos aquellos que tengan un valor mayor o 21, si la casa se pasa de 21, todos los jugadores ganan. (Si el jugador se pasa de esos 21 puntos pierde, indistintamente de lo que haga la casa)

1. Suma con dados
2. Materiales: 3 dados, hojas para escribir, lápiz
3. Duración: 15-20 min
4. Instrucciones:

i. La actividad tiene como objetivo introducir el funcionamiento de un loop.

ii.**While.** Se toma un par de dados y se tiran tres veces, el resultado (x) de la suma será la condición inicial del while. Se anotará este número. (While < x)

iii.Después, se toma un solo dado y se lanzará. El número que salga será el primer valor que se tomará en cuenta, por lo que deberá de anotarse.

iv.Se lanzará el dado y se sumará en cada lanzamiento el número que cae,

Por ejemplo, se repite este proceso siempre y cuando el resultado de la suma cumpla con la condición inicial. **Ejemplo:** Si en el punto ii) la condición inicial fue 36, se tirará el dado n número de veces y se sumará cada número, siempre y cuando el resultado de la suma sea menor a 36.

vi.**For.** Se toma un dado, se lanza y el valor que salga será el valor inicial(x). De manera similar, se toman dos dados y se lanzan dos veces. El resultado de la suma de los dos lanzamientos será el valor final (y). El intervalo de valores es [x , y].

vii. Posteriormente, se tomará un dado y se lanzará. El valor de este lanzamiento se convertirá en el contador para el for. Esto quiere decir, que al valor inicial se le sumará el contador, n número de veces hasta llegar al valor final.

viii. Se escribirá una lista con el valor inicial, final, y todos los números dentro de este intervalo que sean parte de la suma.

Actividades de Programación:

Del 1 al 10

1. Crea un archivo y nómbralo while10.py

Este programa imprimirá los números del uno al 10.

1. Crea tu variable contador y asígnale el valor de 1.
2. Haz un loop utilizando while.

(Mientras el contador sea menor o igual a 10, se imprime el contador.)

¿Cómo le puedes hacer para que el contador sea una unidad mayor cada ciclo?

Del 1 al 10 en 2 líneas

1. Crea un archivo y nómbralo For10.py

Haz un programa que cuente del 1 al 10. Esta vez no necesitaremos más de 2 líneas. ¿No me crees?

1. Escribe:

for i in range (1,10):

print i

1. Corre el programa. ¿Qué pasó?
2. Sólo imprime hasta el 9 porque los rangos no incluyen el último número. Por eso debemos especificar que sea del 1 al 11.
3. Córrelo otra vez.
4. Así de fácil.

**Todos los números en un rango**

1. Crea un archivo y nómbralo for\_numeros.py

Objetivo: Crear un programa que imprima todos los números enteros dentro de un rango de números. **Ejemplo: del 1 al 10, del 5 al 25, del 4 al 1500.**

1. Crea dos variables int, una para el número de inicio y otra para el número final, ambos valores serán ingresados por el usuario.

inicio=int(input('Número inicial: '))

fin=int(input('Número final: '))

1. Tomando en cuenta las consideraciones para el ‘for’ del ejercicio anterior, crea un for para que se impriman todos los números del rango.

for i in range (inicio,fin+1):

print(i)

1. Corre el programa.

**While con Dados**

1. Crea un archivo y nómbralo whiledados.py

Objetivo: Hacer un programa en el que se automaticen las operaciones de la actividad de suma con dados, de las actividades de en la mañana.

1. Crea las variables para cada lanzamiento, estas deben de ser int, porque son números enteros.

l1=int(input('primer lanzamiento:'))

l2=int(input('segundo lanzamiento:'))

l3=int(input('tercer lanzamiento:'))

1. La suma de los lanzamientos será la condición inicial de while, en la que un número tiene que ser menor que la condición establecida. También hay que asignarle un valor de 0 a la variable en la que se guardará la suma de los dados.

lim=l1+l2+l3

count=0

while count<lim:

1. Ahora, falta escribir el proceso que se llevará a cabo en el while, siempre y cuando se cumpla la condición inicial. Recuerda que se debe de registrar el número que salga en el dado y se tienen que estar sumando con cada lanzamiento.

dado=int(input('número: '))

count+=dado

print('el número acumulado es: ', count)

1. Corre el programa. ¿Qué pasó? Discute el resultado con tus compañeras e instructoras.

**For con dados**

1. Crea un programa y nómbralo for\_dados.py

Objetivo: Automatizar la actividad de los dados para el loop for.

1. Crea las variables para los límites inferior y superior del rango de números, estas deben de ser int, porque son números enteros

x=int(input('Lanza un dado y escribe el valor del número que cae: '))

y1=int(input('Lanza dos dados y escribe el valor del número que cae: '))

y2=int(input('Lanza de nuevo dos dados y escribe el valor del número que cae: '))

y=y1+y2

1. Crea la variable para el contador, esta también tiene que ser int.

contador=int(input('Lanza un dado y escribe el número que cae, este será el contador: '))

print('Los números dentro del rango son: ')

1. Ahora es tiempo de escribir el loop

for i in range (x,y+1,contador):

print(i)

1. Corre el programa.

Actividades Extra:

1. Juegos de computadora con python:

Preparar con anticipación las sesiones de juego para que las alumnas puedan iniciar sesión cómodamente: <https://hourofcode.com/us/es/learn>

1. Matamoscas
   1. Duración: 20-30 mins
   2. Actividad grupal
   3. Instrucciones:
      1. Toda la unidad se ubica en un extremo de un terreno previamente delimitado, de aproximadamente 20 x 10 M.
      2. El responsable de la actividad nombra a un participante que se ubicará en el medio del terreno, a su indicación todos los participantes deberán correr arrancando del participante que estaba en el medio.
      3. Los participantes que sean atrapados por éste deberán tomarse de las manos y, sin soltarse, tratarán de atrapar a los otros jugadores que arrancan por el terreno.
      4. Ganará el jugador que sea el último en ser atrapado.