



Prifysgol Cymru
Y Drindod Dewi Sant
University of Wales
Trinity Saint David

ACCA5036

Scripting and Programming Concepts

Lecture 3: Control Structures: **Loops**

Trawsnewid Addysg; Trawsnewid Bywydau
Transforming Education; Transforming Lives



www.uwtsd.ac.uk

Lesson Objectives

- Recap
- Homework solutions
- Understand and use loops to repeat actions.
- *Topics:*
 1. Loops: (i) **While** loops and (ii) **For** loops.
 2. **Nested** Loops
 3. Loop control statements (**break, continue**)

Open Thonny IDE



Debugging – Fix the code

- Objective:

Identify and fix incorrect variable declarations.

```
my-name = "Emma"  
2age = 25  
favorite food = "Pizza"  
isPythonFun = "yes"  
  
print(My-name, "is", 2age, "years old and loves", favorite food)
```

Debugging – Fixed code

- Objective:

Identify and fix incorrect variable declarations.

```
my_name = "Emma"  
age2 = 25  
favorite_food = "Pizza"  
is_python_fun = "yes"  
  
print(my_name, "is", age2, "years old and loves", favorite_food)
```

Debugging – Fix the code

- Objective:

Fix errors related to data types and type conversion.

```
length = "50"  
width = 30.5  
area = length * width # Calculate area  
  
print("The area of the rectangle is " + area)
```

Debugging – Fixed code

- Objective:

Fix errors related to data types and type conversion.

```
length = int("50")  # Convert to integer
width = 30.5
area = length * width

print(f"The area of the rectangle is {area}")
```

Debugging – Fix the code

- Objective:

Fix errors related to taking input from users.

```
name = input(Enter your name: )  
age = input("Enter your age: ")  
print("Hello, " name "! You are" age "years old.")
```

Debugging – Fixed code

- Objective:

Fix errors related to taking input from users.

```
name = input("Enter your name: ")
age = input("Enter your age: ")

print("Hello, " + name + "! You are " + age + " years old.")
```

Debugging – Fix the code

- Objective:

Fix incorrect mathematical operations.

```
num1 = input("Enter first number: ")
num2 = input("Enter second number: ")
sum = num1 + num2
print("The sum of the numbers is:", sum)
```

Debugging – Fixed code

- Objective:

Fix incorrect mathematical operations.

```
num1 = int(input("Enter first number: "))  
num2 = int(input("Enter second number: "))  
total = num1 + num2  
  
print("The sum of the numbers is:", total)
```

Debugging – Fix the code

- Objective:

Fix errors related to printing formatted output.

```
name = input("Enter your name: ")
age = input("Enter your age: ")

print("Hello {name}, you are {age} years old!")
```

Debugging – Fixed code

- Objective:

Fix errors related to printing formatted output.

```
name = input("Enter your name: ")
age = input("Enter your age: ")

print("Hello {name}, you are {age} years old!")
```

```
print(f"Hello {name}, you are {age} years old!")
```

Modify the program to **ask for their city** and include it in the output.

Debugging – Fix the code

```
name = input("Enter your name: ")
age = input("Enter your age: ")
hobby = input("Enter your hobby: ")

years_until_100 = 100 - age
print(f"{name}, you will turn 100 in {years_until_100} years!")
```

Can you find the bug?

Recap - Quiz

- What will be the output of the following code if the user enters John?

```
name = input("Enter your name: ")  
print("Hello", name)
```

- A) Enter your name: John
Hello John
- B) Hello name
- C) Enter your name: John
- D) Error

Recap - Quiz

- What will be the output if the user enters 10?

```
x = int(input("Enter a number: "))  
  
if x > 5:  
    print("Greater than 5")
```

- A) Enter a number: 10
Greater than 5
- B) Enter a number: 10
- C) Greater than 5
- D) Error

Recap - Quiz

- What will be the output if the user enters 3?

```
num = int(input("Enter a number: "))  
if num % 2 == 0:  
    print("Even")  
else:  
    print("Odd")
```

- A) Even
- B) Odd
- C) Error
- D) Nothing

Recap - Quiz

- What will be the output if the user enters 90?

```
marks = int(input("Enter your marks: "))  
if marks >= 90:  
    print("Grade A")  
elif marks >= 75:  
    print("Grade B")  
else:  
    print("Grade C")
```

- A) Grade A
- B) Grade B
- C) Grade C
- D) Error

Recap - Quiz

- What will be the output if the user enters 50?

```
x = int(input("Enter a number: "))  
if x > 0:  
    if x < 100:  
        print("Positive and less than 100")
```

- A) Positive and less than 100
- B) Nothing
- C) Error
- D) Positive

Recap - Quiz

- What will be the output if the user enters 10?

```
x = input("Enter a number: ")
if x == 10:
    print("Equal to 10")
else:
    print("Not equal to 10")
```

- A) Equal to 10
- B) Not equal to 10
- C) Error
- D) None of the above

Recap - Quiz

- What will be the output if the user enters 5?

```
x = input("Enter a number: ")
if x > 3:
    print("Greater than 3")
else:
    print("Less than or equal to 3")
```

- A) Greater than 3
- B) Less than or equal to 3
- C) Error
- D) None of the above

Recap - Quiz

- What happens when the following code runs?

```
num = int(input("Enter a number: "))  
if num > 0:  
    print("Positive")  
else:  
    print("Negative or Zero")
```

- A)** Positive
- B)** Negative or Zero
- C)** Indentation Error
- D)** None of the above

Recap - Quiz

- What will be the output if the user enters 15?

```
1 x = int(input("Enter a number: "))
2 if x > 10 and x < 20:
3     print("Between 10 and 20")
4 elif x > 14:
5     print("Greater than 20")
6 else:
7     print("10 or less")
```

- A)** Between 10 and 20
- B)** Greater than 20
- C)** 10 or less
- D)** Error

Recap - Quiz

- What will be the output if the user enters 80?

```
marks = int(input("Enter marks: "))  
if marks >= 60:  
    print("Passed")  
elif marks >= 80:  
    print("Excellent")  
else:  
    print("Failed")
```

- A)** Passed
- B)** Excellent
- C)** Passed and Excellent
- D)** Failed

Recap - Quiz

- What will be the output if the user enters 0?

```
num = int(input("Enter a number: "))  
  
if num:  
    print("Non-zero")  
  
else:  
    print("Zero")
```

- A)** Non-zero
- B)** Zero
- C)** Error
- D)** None of the above

I was trying to do it
online, but AI deleted
it.



Activity-Age Classification Program

Write a program that asks for a user's age and categorizes them into different age groups.

Instructions:

Input the user's age.

If the age is less than 13, display "You are a child."

If the age is between 13 and 19 (inclusive), display "You are a teenager."

If the age is between 20 and 60 (inclusive), display "You are an adult."

If the age is above 60, display "You are a senior."

Activity-Age Classification Program

```
age = int(input("Enter your age: "))

if age < 13:
    print("You are a child.")
elif 13 <= age <= 19:
    print("You are a teenager.")
elif 20 <= age <= 60:
    print("You are an adult.")
else:
    print("You are a senior.")
```

Activity-Age Classification Program

Write a program that asks for a user's age and categorizes them into different age groups.

Instructions:

Input the user's age.

If the age is less than 13, display "You are a child."

If the age is between 13 and 19 (inclusive), display "You are a teenager."

If the age is between 20 and 60 (inclusive), display "You are an adult."

If the age is above 60, display "You are a senior."

Modify the program to check if the age is less than '0' to print 'this is an invalid age'

Activity-Age Classification Program

Modify the program to check if the age is less than '0' to print 'this is an invalid age'

```
age = int(input("Enter your age: "))

if age < 0:
    print("This is an invalid age.")
elif age < 13:
    print("You are a child.")
elif 13 <= age <= 19:
    print("You are a teenager.")
elif 20 <= age <= 60:
    print("You are an adult.")
else:
    print("You are a senior.")
```

Activity-Age Classification Program

Modify the program to check if the age is less than '0' and greater than '105' to print 'this is an invalid age'

if age < 0 or age > 105 :

```
age = int(input("Enter your age: "))

if age < 0:
    print("This is an invalid age.")
elif age < 13:
    print("You are a child.")
elif 13 <= age <= 19:
    print("You are a teenager.")
elif 20 <= age <= 60:
    print("You are an adult.")
else:
    print("You are a senior.")
```

Activity-Even or Odd Number Checker

Write a program to check if a given number is even or odd.

Instructions:

Input a number from the user. If the number is divisible by 2 (i.e., number $\% 2 == 0$), print "Even."

If the number is not divisible by 2 (i.e., number $\% 2 \neq 0$), print "Odd."

Activity-Even or Odd Number Checker

```
# Input a number from the user  
num = int(input("Enter a number: "))  
  
# Check if the number is even or odd  
if num % 2 == 0:  
    print("Even")  
else:  
    print("Odd")
```

Activity-Voting Eligibility Program

Write a program that checks if a user is eligible to vote.

Instructions:

Input the user's age.

If the age is 18 or older, print "You are eligible to vote."

If the age is less than 18, print "You are not eligible to vote yet."

Activity-Voting Eligibility Program

```
age = int(input("Enter your age: "))

# Check eligibility to vote

if age >= 18:
    print("You are eligible to vote.")
else:
    print("You are not eligible to vote yet.")
```

Activity-Number Comparison Program

Write a program that takes two numbers as input and compares them.

Instructions:

Input two numbers from the user.

If the first number is greater than the second, print "The first number is greater."

If the first number is smaller than the second, print "The second number is greater."

If the two numbers are equal, print "The numbers are equal."

Activity-Number Comparison Program

```
num1 = float(input("Enter the first number: "))
num2 = float(input("Enter the second number: "))

# Compare the numbers

if num1 > num2:
    print("The first number is greater.")
elif num1 < num2:
    print("The second number is greater.")
else:
    print("The numbers are equal.")
```

Activity-Simple Calculator Program

Create a simple calculator that performs basic operations.

Instructions:

Input two numbers from the user.

Ask the user to choose an operation: addition, subtraction, multiplication, or division.

If the operation is addition, print the sum.

If the operation is subtraction, print the difference.

If the operation is multiplication, print the product.

If the operation is division, print the quotient and remainder.

Activity-Simple Calculator Program

```
# Input two numbers from the user
num1 = float(input("Enter the first number: "))
num2 = float(input("Enter the second number: "))

# Display operation choices
print("Choose an operation:")
print("1. Addition (+)")
print("2. Subtraction (-)")
print("3. Multiplication (*)")
print("4. Division (/)")

# Take the user's choice
choice = input("Enter the operation (+, -, *, /): ")
```

```
# Perform the chosen operation
if choice == "+":
    print(f"The sum is: {num1 + num2}")
elif choice == "-":
    print(f"The difference is: {num1 - num2}")
elif choice == "*":
    print(f"The product is: {num1 * num2}")
elif choice == "/":
    if num2 != 0:
        quotient = num1 / num2
        remainder = num1 % num2
        print(f"The quotient is: {quotient}")
        print(f"The remainder is: {remainder}")
    else:
        print("Error! Division by zero is not allowed.")
else:
    print("Invalid operation. Please enter +, -, *, or /.")
```

Activity-Grade Evaluation Program

Write a program to assign a grade based on the marks input by the user.

Instructions:

Input the marks from the user.

If the marks are above or equal to 90, assign grade "A."

If the marks are between 70 and 89, assign grade "B."

If the marks are between 50 and 69, assign grade "C."

If the marks are below 50, assign grade "F."

Activity-Grade Evaluation Program

```
marks = int(input("Enter your marks: "))

# Determine the grade based on the marks
if marks >= 90:
    grade = "A"
elif marks >= 70:
    grade = "B"
elif marks >= 50:
    grade = "C"
else:
    grade = "F"

# Output the grade
print("Your grade is:", grade)
```

Activity-Display sequence of number

Write a program to print the sequence of numbers up to n.

Instructions:

Input the number n between 1 and 5 from the user.

If the number is 1, print 1.

If the number is 2, print 1,2.

If the number is 3, print 1,2,3.

If the number is 4, print 1,2,3,4.

If the number is 5, print 1,2,3,4,5.

```
Enter a number: 5
```

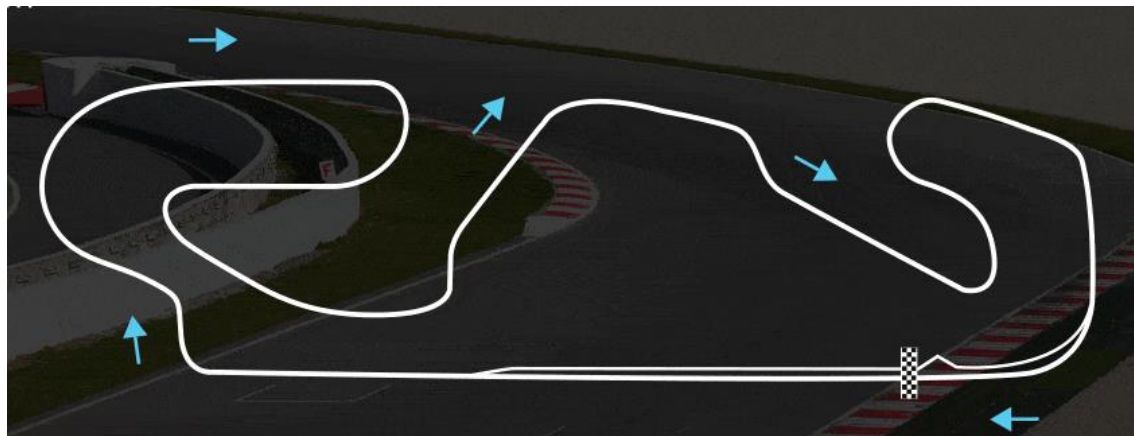
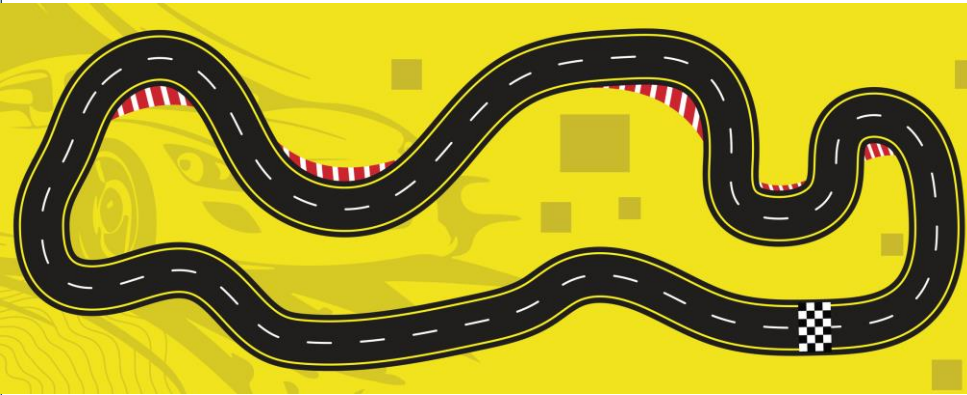
```
1 2 3 4 5
```

Activity-Display sequence of number

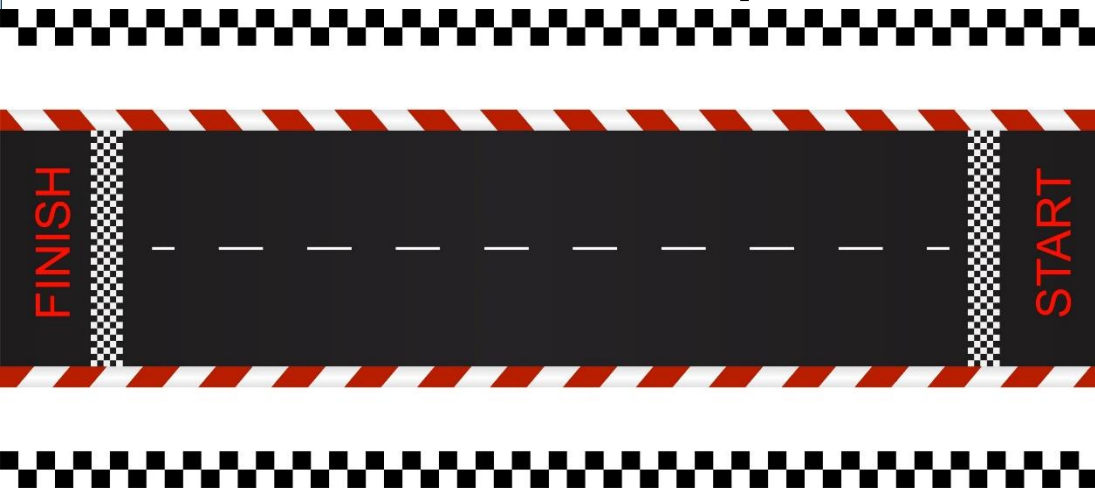
```
# Input a number from the user
n = int(input("Enter a number between 1 and 5: "))

# Check and print the sequence based on the input
if n == 1:
    print("1")
elif n == 2:
    print("1,2")
elif n == 3:
    print("1,2,3")
elif n == 4:
    print("1,2,3,4")
elif n == 5:
    print("1,2,3,4,5")
else:
    print("Invalid input! Please enter a number between 1 and 5.")
```

Loop: Yes or NO?



Loop: Yes or NO?



Loops in Python

- **Programming with Loops:** In programming, loops are used to *repeat sections* of code based on specified *conditions*.
- **Iterations in Loops:** Loops execute a *set number of times*, referred to as iterations (dictates how many times the loop will run).
- **Types of Loops in Python:** two primary types of loops:
 1. **`for`** loops
 2. **`while`** loops

While loop

While the following condition is true, keep doing the following

```
while condition:  
    statement(s)
```

```
while condition:  
    # Code to execute
```

If the **condition** evaluates to TRUE, then the statements will be executed

While loop - example

```
counter = 1  
# Start a while loop  
while counter <= 5:  
    print(counter)  
    counter = counter + 1
```

While loop - example

```
counter = 1  
# Start a while loop  
while counter <= 5:  
    print(counter)  
    counter = counter + 1
```

While loop - example

```
counter = 0
```

```
# Start a while loop
```

```
while counter <= 5:
```

```
    print(counter)
```

```
    counter = counter + 1
```

While loop - example

```
counter = 0
# Start a while loop
while counter <= 5:
    print(counter)
    counter = counter - 1
```

While loop - example

```
counter = 0
```

```
# Start a while loop
```

```
while counter <= 5:
```

```
    print(counter)
```

```
    counter = counter - 1
```

```
if counter == -5:
```

```
    print("Loop stopped after 5 iterations.")
```

```
    break # Exit the infinite loop
```

while True - example

```
while True:  
    user_input = input("Enter something (type 'quit' to stop): ")
```

while True - example

```
while True:
    user_input = input("Enter something (type 'quit' to stop): ")

    if user_input.lower() == "quit":
        print("You chose to quit. Exiting the loop.")
        break # Exit the loop when user types 'quit'
```

while True - example

```
while True:
    user_input = input("Enter something (type 'quit' to stop): ")

    if user_input.lower() == "quit":
        print("You chose to quit. Exiting the loop.")
        break # Exit the loop when user types 'quit'

    print(f"You entered: {user_input}")
```

Can we use `while not True`?

```
while not True:
```

```
    user_input = input("Enter something (type 'quit' to stop): ")
```

```
    if user_input.lower() == "quit":
```

```
        print("You chose to quit. Exiting the loop.")
```

```
        break # Exit the loop when user types 'quit'
```

```
print(f"You entered: {user_input}")
```

Restaurant Order Program

Enter your **name**: John Doe
Enter your phone number: **123-456-7890**
Enter your address: **123** Main St, Springfield

--- *Restaurant Menu* ---

1. Burger: **£5.00**
2. Pizza: **£8.00**
3. Pasta: **£7.50**
4. Salad: **£4.00**
5. Soda: **£1.50**

Enter the item you want **to order** (or **'done'** to finish): Pizza
How many Pizzas would you **like to order**? **2**
2 Pizza(s) added **to your order**.

Enter the item you want **to order** (or **'done'** to finish): Soda
How many Sodas would you **like to order**? **3**
3 Soda(s) added **to your order**.

Enter the item you want **to order** (or **'done'** to finish): done

Your total **order cost is**: **£22.50**

Thank you **for your order**!

Your **order** will be delivered **in 25 minutes to** the address: **123** Main St, Springfield

Code along...

```
customer_name = input("Enter your name: ")
phone_number = input("Enter your phone number: ")
address = input("Enter your address: ")

# Menu items and prices using string and numeric data types
item1 = "Burger"
price1 = 5.00

item2 = "Pizza"
price2 = 8.00

item3 = "Pasta"
price3 = 7.50

item4 = "Salad"
price4 = 4.00

item5 = "Soda"
price5 = 1.50

# Initialize total cost
total_cost = 0
```



Code along...

```
# Start taking orders
```

```
while True:
```

```
    # Display the menu
```

```
    print("\n--- Restaurant Menu ---")
```

```
    print(f"1. {item1}: £{price1:.2f}")
```

```
    print(f"2. {item2}: £{price2:.2f}")
```

```
    print(f"3. {item3}: £{price3:.2f}")
```

```
    print(f"4. {item4}: £{price4:.2f}")
```

```
    print(f"5. {item5}: £{price5:.2f}")
```

```
    # Take the user's order
```

```
    order_item = input("\nEnter the item order (or 'done' to finish): ").capitalize()
```

```
    if order_item == "Done":
```

```
        print(f"\nYour total order cost is: £{total_cost:.2f}")
```

```
        print("Thank you for your order!")
```

```
        print(f"Your order will be delivered in 25 minutes to the address: {address}")
```

```
        break # End the loop and finish the order
```

Code along...

```
# Check the order and add the appropriate cost based on the selection
if order_item == item1:
    quantity = int(input(f"How many {item1}s would you like to order? "))
    if quantity > 0:
        total_cost += price1 * quantity
        print(f"{quantity} {item1}(s) added to your order.")
    else:
        print("Please order at least 1 item.")
```

Code along...

```
elif order_item == item2:
    quantity = int(input(f"How many {item2}s would you like to order? "))
    if quantity > 0:
        total_cost += price2 * quantity
        print(f"{quantity} {item2}(s) added to your order.")
    else:
        print("Please order at least 1 item.")

elif order_item == item3:
    quantity = int(input(f"How many {item3}s would you like to order? "))
    if quantity > 0:
        total_cost += price3 * quantity
        print(f"{quantity} {item3}(s) added to your order.")
    else:
        print("Please order at least 1 item.")
```

Code along...

```
elif order_item == item5:
    quantity = int(input(f"How many {item5}s would you like to order? "))
    if quantity > 0:
        total_cost += price5 * quantity
        print(f"{quantity} {item5}(s) added to your order.")
    else:
        print("Please order at least 1 item.")

else:
    print("Sorry, that item is not on the menu. Please select a valid item.")
```

While Loop - Example

TRY

Activity:

1. Write and run the following example.
2. Try changing the condition. The current condition is (**turns < 5**)

```
count = 0
while count < 5:
    print(count)
    count += 1
```

Example: a game that allows a set number of turns

```
turns = 0                # initial value
while (turns < 5):
    print("Turns used: ", turns)
    turns += 1            #add 1 to turns
print("You have now used all your turns.")
```

For Loop

TRY
THIS

```
for i in range(5):  
    print(i)
```

```
for i in range(1, 6):  
    print(i)
```

```
for i in range(2, 6):  
    print(i)
```

```
for i in range(100, 110):  
    print(i)
```

```
for i in range(1, 10, 2):  
    print(i)
```

```
for i in range(10, 0, -2):  
    print(i)
```

```
for i in range(11, 0, -2):  
    print(i)
```

```
for i in range(-5, 0):  
    print(i)
```

```
for a in range(3):  
    print(a)  
    print("Hello!")
```

Nested Loops

Outer loop (for numbers 1 to 3)

for i in range(1, 4):

Inner loop (for numbers 1 to 3)

for j in range(1, 4):

print(f"{i} + {j} = {i + j}", end="\t") # Print addition result

print() # Print a newline after each row

Printing patterns

Number of stars to print

n = 5

Print stars in a vertical column

for i in range(n):

print("*")

Printing patterns

```
char = input("Enter the character you want to print: ")  
n = int(input("Enter the number of rows: "))  
for i in range(n):  
    print(char)
```

Printing patterns

```
char = input("Enter the character you want to print: ")
n = int(input("Enter the number of rows: "))
for i in range(n):
    print(char, end=" ")
```

Printing patterns

```
# Get character and number of rows/columns from the user
char = input("Enter the character you want to print: ")
n = int(input("Enter the number of rows/columns: "))
```

```
# Outer loop for rows
for i in range(n):
    # Inner loop for columns
    for j in range(n):
        print(char, end=" ")
    print() # Print a new line after each row
```

Printing patterns

```
char = input("Enter the character you want to print: ")
n = int(input("Enter the number of rows: "))
for i in range(1, n + 1):
    # Inner loop for columns (print i characters in each row)
    for j in range(i):
        print(char, end=" ")
    print() # Print a new line after each row
```

Number Pyramid Pattern

Number of rows

n = 5

Outer loop for rows

for i in range(1, n + 1):

 # Print leading spaces

 for j in range(n - i):

 print(" ", end=" ")

 # Print numbers

 for j in range(1, i + 1):

 print(j, end=" ")

print() # Print a new line after each row



Prifysgol Cymru
Y Drindod Dewi Sant
University of Wales
Trinity Saint David

Trawsnewid Addysg; Trawsnewid Bywydau
Transforming Education; Transforming Lives

```
import random
```

```
# Generate a random number between 1 and 10  
secret_number = random.randint(1, 10)
```

```
# Give the user 5 attempts to guess the number  
for attempt in range(1, 6):
```

```
    guess = int(input(f"Attempt {attempt}/5: Guess the number between 1 and 10: "))
```

```
    if guess == secret_number:
```

```
        print("Congratulations! You guessed the right number!")
```

```
        break
```

```
    else:
```

```
        print("Wrong guess. Try again!")
```

```
else:
```

```
    print(f"Sorry, you've run out of attempts. The correct number was {secret_number}.")
```

For loop

Repeat a procedure a number of times.

marking student work:

→ *apply a procedure* on **all** the answer scripts.



Put shoes in boxes.



How many times is the procedure applied?

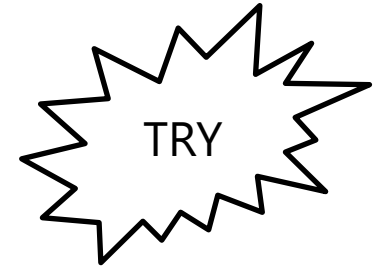
Depends on the items within

Iterate over a set (e.g., list of numbers)

A list of numbers

6	5	77	22	1	7	9	12
---	---	----	----	---	---	---	----

For in lists (1/2)



Example:

```
numbers = [10, 2, 3, 40, 5]
```

Explanation:



'numbers' = List of numbers

```
for num in numbers:  
    print(num)
```



'num' changes as it goes through each item (iterates) – it receives the value of each time. Instead of num you can use any variable name (e.g., x, y, number no, etc.)

Output:

```
10  
2  
3  
40  
5
```

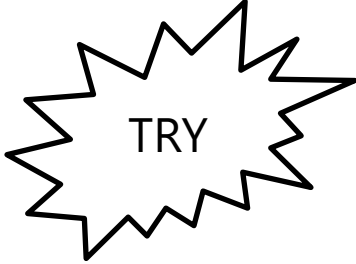
For in lists (2/2)

```
colours=['red', 'blue', 'pink', 'white']
```

```
for i in colours:  
    print (i)
```

output:

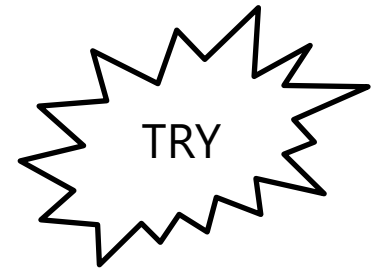
Red
Blue
Pink
white



TRY

```
text = "Hello, world!"  
for char in text:  
    print(char)
```

For Loop -- Range



You can iterate through a **range of numbers**, instead of a list.

```
>>> range(5)
range(0, 5)
```

```
for i in range(5):
    print(i)
```

Output: _____

0
1
2
3
4

```
range(0, 5)
```

```
for i in range(0,5):
    print(i)
```

Output: _____

0
1
2
3
4

```
range(5, 10)
```

```
for i in range(5,10):
    print(i)
```

Output: _____

5
6
7
8
9

The **i** initiated to 0 and will continue the looping until **i** becomes 5 (does not print 5).

For Loop – Range & Step

```
for i in range (1,10,2):  
    print(i)
```

→ step value

The step value 2 will increment the i value by 2.

Output:

1
3
5
7
9

TRY

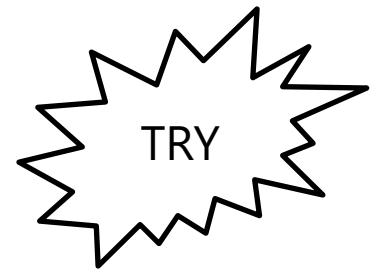
Nested Loops (Loop in a Loop)

You can have a loop in a loop.

```
for i in range(3):  
    for j in range(3):  
        print("outer: ", i, "; inter: ", j)
```

Output:

```
outer: 0 ; inter: 0  
outer: 0 ; inter: 1  
outer: 0 ; inter: 2  
outer: 1 ; inter: 0  
outer: 1 ; inter: 1  
outer: 1 ; inter: 2  
outer: 2 ; inter: 0  
outer: 2 ; inter: 1  
outer: 2 ; inter: 2
```



Avoid Infinite Loops (While Loop)

An infinite loop is a loop that continues to execute indefinitely because its terminating condition is never met.

The following will cause an infinite loop. Why?

```
counterx = 0
while (counterx == 0):
    print("Current number is: ", counterx)
```

`counterx` does not change. The condition (`counterx == 0`): will always be True

```
counterx = 1
while (counterx > 0):
    print("Current number is: ", counterx)
    counterx += 1
```

`counterx` changes but the condition (`counterx > 0`) will always be True.

Controlling Loops - **Break**

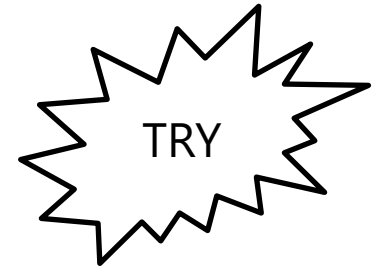
- You can change the default behaviour of a loop.

Break command exits the loop

```
for i in range(5):  
    if i == 3:  
        break  
    print(i)
```

Output:

0
1
2



In this example, the loop stops (exits) when (**i == 3**)

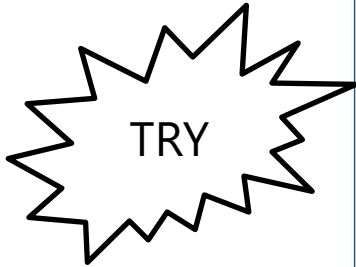
Controlling Loops - **continue**

continue command skips the current iteration and continues.

```
for i in range(5):  
    if i == 3:  
        continue  
    print(i)
```

Output:

0
1
2
4



TRY

In this example, the loop skips when (**i == 3**) and continues with the next step.

<https://forms.office.com/e/j5TZ3K8gNP>



List Datatype

- a **list** is a built-in data structure
- to store multiple items in a single variable
- can hold different types of data, including numbers, strings, and even other lists
- They are ordered, mutable (changeable), and allow duplicate elements.
- You can create a list by placing values inside square brackets [], separated by commas.

Creating a List

```
my_list = [1, 2, 3, 4, 5]  
print(my_list)
```

```
my_list = [5, 2, 3, 4, 5]  
print(my_list)
```

Creating a List

```
my_list = ["apple", "banana", "cherry"]  
print(my_list)
```

```
mixed_list = [1, "hello", 3.14, True]  
print(mixed_list)
```

Creating a List

```
mixed_list = [1, "hello", 3.14, True, ["apple",  
"banana", "cherry"]]  
print(mixed_list)
```

List Operations

Accessing Elements

```
fruits = ["apple", "banana", "cherry"]  
print(fruits[0]) # Access the first element
```

```
fruits = ["apple", "banana", "cherry"]  
print(fruits[-1]) # Access the last element
```

List Operations

List Slicing

```
fruits = ["apple", "banana", "cherry"]  
print(fruits[1:2])
```

```
numbers = [1, 2, 3, 4, 5]  
print(numbers[1:4])
```

List Operations

Modifying a List

```
fruits = ["apple", "banana", "cherry"]  
fruits[1] = "blueberry"  
print(fruits)
```

```
fruits = ["apple", "banana", "cherry"]  
fruits.append("date")  
print(fruits)
```

List Operations

Modifying a List

```
fruits = ["apple", "banana", "cherry"]  
fruits.insert(1, "blueberry") # Insert 'blueberry' at index 1  
print(fruits)
```

```
fruits = ["apple", "banana", "cherry"]  
fruits.extend(["date", "elderberry"])  
print(fruits)
```

List Operations

Modifying a List

```
fruits = ["apple", "banana", "cherry"]  
fruits.remove("banana")  
print(fruits)
```

```
fruits = ["apple", "banana", "cherry"]  
del fruits[1] # Delete the item at index 1  
print(fruits)
```

```
fruits = ["apple", "banana", "cherry"]  
popped_item = fruits.pop(1)  
print(fruits)  
print("Popped item:", popped_item)
```

Summary

Key Points:

- Loops are essential for repeating actions in code.
- *While loops* continue as long as a condition is true.
- *For loops* iterate over a sequence of items.
- *Nested loops* are useful for multi-dimensional problems.
- Loop control statements (*break*, *continue*) alter the flow of loops.