
PSEUDOCODES
& ACTIVITIES

THINGS TO DO TODAY

GRAD 10 STUDENTS

You are the log-in to HTTPS://IDEA.Org.Uk, and get the bronze, silver, or awards by completing ICT/CS related badges

IGCSE CS 0478 CANDIDATES

Take the front row and pay attention with the activities for PSEUDOCODEs.

LOOPS IN PSEUDOCODES

Loops are used to repeat a set of instructions some number of times. This is a programming concept known as iteration or repetition.

In the python programming language, there are two different loop structures that we can use. One structure is a for loop, and the other structure is a while loop.

When writing pseudocode, we can still use **FOR loops and WHILE loops**, but there is also a third loop structure using the **REPEAT command**.

USING
THE WHILE LOOP

For loops

Example 1

```
x = 1
while x <= 10:
    print(x)
    x = x + 1
```

Example 2

```
Sum = 0
while Sum < 100:
    x = int(input())
    Sum = Sum + x
print(Sum)
```

Example 3

```
Sum = 0
x = input()
while x != 0:
    Sum = Sum + x
    x = int(input())
print(Sum)
```

For loops

Example 1

```
x = 1
while x <= 10:
    print(x)
    x = x + 1
```

Example 2

```
Sum = 0
while Sum < 100:
    x = int(input())
    Sum = Sum + x
print(Sum)
```

Example 3

```
Sum = 0
x = input()
while x != 0:
    Sum = Sum + x
    x = int(input())
print(Sum)
```

Example 1

```
X ← 1
WHILE X <= 10 DO
    PRINT X
    X ← X + 1
ENDWHILE
```

For loops

Example 1

```
x = 1
while x <= 10:
    print(x)
    x = x + 1
```

Example 2

```
Sum = 0
while Sum < 100:
    x = int(input())
    Sum = Sum + x
print(Sum)
```

Example 3

```
Sum = 0
x = input()
while x != 0:
    Sum = Sum + x
    x = int(input())
print(Sum)
```

Example 1

```
X ← 1
WHILE X <= 10 DO
    PRINT X
    X ← X + 1
ENDWHILE
```

Example 2

```
Sum ← 0
WHILE Sum < 100 DO
    INPUT X
    Sum ← Sum + X
ENDWHILE
PRINT Sum
```

For loops

Example 1

```
x = 1
while x <= 10:
    print(x)
    x = x + 1
```

Example 2

```
Sum = 0
while Sum < 100:
    x = int(input())
    Sum = Sum + x
print(Sum)
```

Example 3

```
Sum = 0
x = input()
while x != 0:
    Sum = Sum + x
    x = int(input())
print(Sum)
```

Example 1

```
X ← 1
WHILE X <= 10 DO
    PRINT X
    X ← X + 1
ENDWHILE
```

Example 2

```
Sum ← 0
WHILE Sum < 100 DO
    INPUT X
    Sum ← Sum + X
ENDWHILE
PRINT Sum
```

Example 3

```
Sum ← 0
INPUT X
WHILE X <> 0 DO
    Sum ← Sum + X
    INPUT X
ENDWHILE
PRINT Sum
```

WHILE loops

```
x = 1
while x <= 5:
    print(x)
    x = x + 1


Count = 0
while Count < 3:
    print(Count)
    Count = Count + 1


Number = int(input())
Power = 0
while Power <= 10:
    print(Number ** Power)
    Power = Power + 1


x = int(input())
Multiple = 0
while Multiple < 1000:
    Multiple = Multiple + x
    print(Multiple)
```

WHILE loops

```
x = 1
while x <= 5:
    print(x)
    x = x + 1
```

```
Count = 0
while Count < 3:
    print(Count)
    Count = Count + 1
```

```
Number = int(input())
Power = 0
while Power <= 10:
    print(Number ** Power)
    Power = Power + 1
```

```
x = int(input())
Multiple = 0
while Multiple < 1000:
    Multiple = Multiple + x
    print(Multiple)
```

(a)
```
x ← 1
WHILE x <= 5 DO
    PRINT x
    x ← x + 1
ENDWHILE
```

(b)
```
Count ← 0
WHILE Count < 3 DO
    PRINT Count
    Count ← Count + 1
ENDWHILE
```

WHILE loops

```
x = 1
while x <= 5:
    print(x)
    x = x + 1
```

```
Count = 0
while Count < 3:
    print(Count)
    Count = Count + 1
```

```
Number = int(input())
Power = 0
while Power <= 10:
    print(Number ** Power)
    Power = Power + 1
```

```
x = int(input())
Multiple = 0
while Multiple < 1000:
    Multiple = Multiple + x
    print(Multiple)
```

(a)
```
x ← 1
WHILE x <= 5 DO
    PRINT x
    x ← x + 1
ENDWHILE
```

(b)
```
Count ← 0
WHILE Count < 3 DO
    PRINT Count
    Count ← Count + 1
ENDWHILE
```

(c)
```
INPUT Number
Power ← 0
WHILE Power <= 10 DO
    PRINT Number ^ Power
    Power ← Power + 1
ENDWHILE
```

(d)
```
INPUT x
Multiple ← 0
WHILE Multiple < 1000 DO
    Multiple ← Multiple + x
    PRINT Multiple
ENDWHILE
```

WHILE loops

```
x = 1
while x <= 5:
    print(x)
    x = x + 1
```

```
Count = 0
while Count < 3:
    print(Count)
    Count = Count + 1
```

```
Number = int(input())
Power = 0
while Power <= 10:
    print(Number ** Power)
    Power = Power + 1
```

```
x = int(input())
Multiple = 0
while Multiple < 1000:
    Multiple = Multiple + x
    print(Multiple)
```

(a)
```
x ← 1
WHILE x <= 5 DO
    PRINT x
    x ← x + 1
ENDWHILE
```

(b)
```
Count ← 0
WHILE Count < 3 DO
    PRINT Count
    Count ← Count + 1
ENDWHILE
```

(c)
```
INPUT Number
Power ← 0
WHILE Power <= 10 DO
    PRINT Number ^ Power
    Power ← Power + 1
ENDWHILE
```

(d)
```
INPUT x
Multiple ← 0
WHILE Multiple < 1000 DO
    Multiple ← Multiple + x
    PRINT Multiple
ENDWHILE
```

(e)
```
INPUT x
Multiple ← x
WHILE Multiple < 1000 DO
    PRINT Multiple
    Multiple ← Multiple + x
ENDWHILE
```

(f)
```
Total ← 0
INPUT Value
WHILE Value >= 0 DO
    Total ← Total + Value
    INPUT Value
ENDWHILE
PRINT Total
```

WHILE loops worded problems

DECLARE ANY VARIABLES, CONSTANTS OR ARRAYS USED BY YOUR ALGORITHM.

1. Write an algorithm, using pseudocode containing a WHILE loop, to print "Hello" five times.

2. Write an algorithm, using pseudocode with a WHILE loop, to print all the numbers 1 to 20.

3. Write an algorithm, using pseudocode containing a `WHILE` loop, to do the following 5 times:

    - ask the user to enter a positive whole number
    - output the value that is the remainder when the number entered is divided by 7

WHILE loops worded problems

DECLARE ANY VARIABLES, CONSTANTS OR ARRAYS USED BY YOUR ALGORITHM.

1. Write an algorithm, using pseudocode containing a WHILE loop, to print "Hello" five times.

2. Write an algorithm, using pseudocode with a WHILE loop, to print all the numbers 1 to 20.

3. Write an algorithm, using pseudocode containing a `WHILE` loop, to do the following 5 times:

   - ask the user to enter a positive whole number
   - output the value that is the remainder when the number entered is divided by 7

```
DECLARE Count : INTEGER
Count ← 1
WHILE Count ≤ 5 DO
      PRINT "Hello"
      Count ← Count + 1
ENDWHILE
```

WHILE loops worded problems

DECLARE ANY VARIABLES, CONSTANTS OR ARRAYS USED BY YOUR ALGORITHM.

1. Write an algorithm, using pseudocode containing a WHILE loop, to print "Hello" five times.

2. Write an algorithm, using pseudocode with a WHILE loop, to print all the numbers 1 to 20.

3. Write an algorithm, using pseudocode containing a `WHILE` loop, to do the following 5 times:

   - ask the user to enter a positive whole number
   - output the value that is the remainder when the number entered is divided by 7

```
DECLARE Count : INTEGER
Count ← 1
WHILE Count ≤ 5 DO
      PRINT "Hello"
      Count ← Count + 1
ENDWHILE
```

```
DECLARE Count : INTEGER
Count ← 1
WHILE Count ≤ 20 DO
      PRINT Count
      Count ← Count + 1
ENDWHILE
```

WHILE loops worded problems

DECLARE ANY VARIABLES, CONSTANTS OR ARRAYS USED BY YOUR ALGORITHM.

1. Write an algorithm, using pseudocode containing a WHILE loop, to print "Hello" five times.

```
DECLARE Count : INTEGER
Count ← 1
WHILE Count ≤ 5 DO
      PRINT "Hello"
      Count ← Count + 1
ENDWHILE
```

2. Write an algorithm, using pseudocode with a WHILE loop, to print all the numbers 1 to 20.

```
DECLARE Count : INTEGER
Count ← 1
WHILE Count ≤ 20 DO
      PRINT Count
      Count ← Count + 1
ENDWHILE
```

3. Write an algorithm, using pseudocode containing a `WHILE` loop, to do the following 5 times:

   - ask the user to enter a positive whole number
   - output the value that is the remainder when the number entered is divided by 7

```
DECLARE Count : INTEGER
DECLARE Num : INTEGER
Count ← 1
WHILE Count ≤ 5 DO
      INPUT Num
      PRINT Num MOD 7
      Count ← Count + 1
ENDWHILE
```

WHILE loops worded problems

DECLARE ANY VARIABLES, CONSTANTS OR ARRAYS USED BY YOUR ALGORITHM.

4. Write an algorithm, using pseudocode containing a `WHILE` loop, to:

   - ask the user to enter heights of people, in metres
   - stop asking for heights when the user enters a value that is not in $0 < height < 2$
   - finds and outputs the average of the heights entered

```
DECLARE Count : INTEGER
Count ← 1
WHILE Count ≤ 5 DO
      PRINT "Hello"
      Count ← Count + 1
ENDWHILE
```

5. Write an algorithm, using pseudocode containing a `WHILE` loop, to:

   - ask the user to enter numbers
   - stop asking for numbers when the user enters zero
   - store all the entered numbers (except the zero) into an array

```
DECLARE Num : REAL
DECLARE Count : INTEGER
DECLARE A : ARRAY [1:1000] OF REAL
INPUT Num
Count ← 1
WHILE Num <> 0 DO
      A[Count] ← Num
      Count ← Count + 1
      INPUT Num
ENDWHILE
```

THANK YOU

CREDIT

Credit to text, illustrations, images, videos, recordings, etc. owners in this presentation.

For educational purposes only.