



Assignment

Assignment is when you make a **variable** become equal to a new value. In Python we use the equals sign = as an **assignment operator**.

Example:

```
myAge = 17
```

This **statement** creates a **variable** called myAge and assigns the value 17 to it.

Arithmetic Operators

We can use arithmetic operators in a similar way to how we would use them in maths.

The operators we can use in Python are:

Name	Symbol	Description	Example
Addition	+	Adds two numbers	1 + 1 = 2
Subtraction	-	Finds the difference between two numbers.	2 - 1 = 1
Multiplication	*	Finds the product of two numbers.	2 * 2 = 4
Division	/	Divides one number by another.	4 / 2 = 2
Modulus	%	Finds the remainder when two numbers are divided.	5 % 2 = 1
Exponent	**	Finds one number multiplied to the power of the second.	3 ** 2 = 9
Floor division	//	Whole number division without decimals, fractions of remainders	5 // 2 = 2

Comments

Comments are used to leave notes for Humans reading the code. They are ignored by the computer. The apostrophe symbol starts a comment.

Example:

```
` The computer will ignore everything  
` on a line after an apostrophe
```

Output

We can output data to the screen for the user to see by using the print **function**.

Example:

```
print("Hello World")
```

To output a **literal** string you enclose it in speech marks. Or you can output the contents of a variable.

```
print(myAge)
```

You can output more than one variable or literal at once by using either commas or plus signs. Using commas will put a space between each item, a plus sign won't.

```
print("You are", myAge, "years old.")
```

```
print("You are " + myAge + " years old.")
```

Input

We can take input from the user and store it in a variable to use later. The example below will ask the user their name, store it in a variable called MyName and then output a personalized message using their name.

```
myName = input("What is your name?")
```

```
print("Hello", myName)
```

This example combines both input and output in a single program. Its possible to take input without storing it in a variable—but then you wouldn't be able to use the input in anyway later in the program.

Iteration

Iteration is when there is a loop in your code so that a block of code is run more than once. There's two types of loop in Python: for-loop and while-loop.

For Loop

Repeats code a set number of times. This example will repeat 4 times:

```
for num1 in range(1 , 5):  
    print (num1)
```

It creates a variable called num1 which equals 1 to start with and after every loop it adds 1 to it until it equals 5—then it stops. This example would print 1, 2, 3, 4.

While loop

A while loop will only run the code in the block while a condition is true. This example will loop until the user types "quit".

```
while ans != "quit":  
    print(ans)
```

Selection

Selection is used to make decisions in a program. We do this by performing a **logical comparison**. If the result of the comparison is **true** then one block of code is executed. If the result is **false** then the code block is skipped.

In Python we can do selection using If statements.

Example:

```
if myAge > 17:  
    print("You are old enough to drive")
```

This example would only output "You are old enough to drive" if myAge is bigger than 17. If it is 17 or less then it won't output anything.

We can also specify a different action if the code is false by using 'else'.

Example:

```
if myAge > 17:  
    print("You are old enough to drive")  
else:  
    print("You cannot drive")
```

We can specify more than one alternative by using the else-if (written in Python as 'elif' for short).

Example:

```
if myAge > 17:  
    print("You are old enough to drive")  
elif myAge == 17:  
    print("You can drive next year!")  
else:  
    print("You cannot drive")
```

Logical Operators

Logical operators are used in comparisons – such as in if statements or while loops.

Operator	Description	Example
==	Equal to	if myAge == 17:
!=	Not equal to	if myAge != 17:
>	Greater than	if myAge > 17:
>=	Greater than or equal to	if myAge >= 17:
<	Less than	if myAge < 17:
<=	Less than or equal to	if myAge <= 17: