

Day 01

MATH 3733 Mathematical Computing

Today's Plan

Set up the learning environment

Review syllabus

Terminology

Operators

Variables

Terminology

What is a "computer"?

What does a computer do?

Performs calculations

Remember calculations

What a human tells it to do

“What I mean is that if you really want to understand something, the best way is to try and explain it to someone else. That forces you to sort it out in your own mind... And that’s really the essence of programming. By the time you’ve sorted out a complicated idea into little steps that even a stupid machine can deal with, you’ve certainly learned something about it yourself.” — Douglas Adams, Dirk Gently's Holistic Detective Agency

”

Computation is any type of calculation that includes both arithmetic and non-arithmetic steps and which follows a well-defined model (e.g., an **algorithm**).

A **program** is a sequence of instructions that specifies how to perform a computation.

A **Read-Eval-Print Loop (REPL)**, also known as an interactive shell, is a simple, interactive computer programming environment that takes single user inputs (READ) (i.e. single expressions), EVALUATES them, and returns (PRINTS) the result to the user; a program written in a REPL environment is executed piecewise.

Official Python Docs: 3.10 Using Python as a Calculator

Arithmetic operators follow the standard order of operation.

- `()` : parentheses
- `**` : exponentiation
- `+` : addition
- `-` : subtraction
- `*` : multiplication
- `/` : division
- `//` : floor division (quotient)
- `%` : modulus (remainder)

Comments

Python supports **comments** via the `#` symbol. You can use the `#` symbol to tell Python to essentially ignore anything else on that line which comes after the symbol. Here are some examples:

```
print('Hello, world!') # this will print the string Hello World!  
  
# the following line will print the string My Name  
print('My Name')
```

Exercise

Write a mathematical expression in a `print` function that satisfies the comments.



```
# Example
# How many seconds are there in 2 minutes, 42 seconds?
print(2 * 60 + 42) # this will print 162

# Exercise 1
# How many seconds are there in 2 hours, 42 minutes, 42 seconds?
print()



# Exercise 2
# Write an expression that simplifies to 42.0 and
# uses the exponentiation, addition, subtraction,
# multiplication, and division operators.
print()
```

Rules for variables



No spaces. Use underscore in place of a space

-  `my var` is not valid
-  `my_var` is valid

The first character of a variable name must be a letter or the underscore character.

-  `4var` is invalid
-  `_message` is valid

No characters are allowed in a variable name besides alphanumeric (alphabet or numeric) characters and the underscore ("_") character (no special characters like &, !, +, \$, etc.)

-  `you&i` is invalid
-  `you_and_i` is valid

Python is case sensitive, so two variables with the same name but different case are two different variables

- `name` , `NAME` , and `Name` are all different variables

✗ You *cannot* use any of Python's built in "reserved words" or **keywords**.

```
>>> import keyword
>>> keyword.kwlist
['False', 'None', 'True', 'and', 'as', 'assert', 'async', 'await', 'break',
'class', 'continue', 'def', 'del', 'elif', 'else', 'except', 'finally',
'for', 'from', 'global', 'if', 'import', 'in', 'is', 'lambda', 'nonlocal',
'not', 'or', 'pass', 'raise', 'return', 'try', 'while', 'with', 'yield']
```

⚠ Do **not** create a variable named `print` because that would conflict with the pre-defined `print` function.

More on **namespace** later.