type()
Determine requirements
↓
Write the source code
↓
Convert source code to object code
↓
Run the program
↓
Check the output
Like all high-level programming languages, Python programs must be compiled and/or interpreted to run. Python scripts have to be processed by another program called the “Python interpreter”.

The interpreter does the following:

- Reads your script
- Compiles it into “bytecode”
- Executes bytecode to run program
Debugging

Syntax errors:
Program doesn’t run because structure isn’t correct or doesn’t follow rules of language

Runtime errors:
Happen when the program is running. Also called “exceptions”

Semantic errors:
Program runs without an error message, but not correctly because it’s not doing what you meant it to
Variable Assignment

Variables are assigned with the assignment token, =

price = 19.99

Assigning more than one variable at a time is possible

a, b, c = 'foo', 'bar', 'baz'

Swapping variable values

x, y = y, x
Basic Data Types

Integers
Floating Point Numbers
Strings
**Integer**

`int`

A whole number

Can be of unlimited size

Be careful not to use commas in your numbers, for example: 1,000
**Floating point number**

*float*

- A number that contains a decimal point
- Has minimum and maximum values
- Limited precision
- 4 is not the same as 4.0
Formatting Numbers

**format()**

The format function allows you to format numbers, like floats.

Takes two arguments: the value to be formatted, and the format specification.

`format(value, 'format_spec')`

The last number will also be rounded as appropriate.
String
str

A sequence of one or more characters

A string of letters and numbers

Enclosed in quotation marks

Double quoted strings can contain single quotes and vice versa

Triple quotes can span multiple lines
String

String Operators

+  *
  #