



CL01: Objects and Data Types

First, an introduction to Visual Studio...

The image shows a screenshot of the Visual Studio IDE interface. The Explorer window on the left is highlighted with a yellow box and labeled 'A Explorer'. The Editor window in the center is highlighted with a cyan box and labeled 'Editor', with a cyan arrow pointing to the text 'B ← Edit a module'. The Terminal window at the bottom is highlighted with a green box and labeled 'Terminal', with a green arrow pointing to the text 'C ← Runs your Command Line Interface (CLI)'. The Terminal window contains the text 'alyssa@macbook-pro-119 comp110-23s-workspace %' and a bulleted list: '• Run a module' and '• REPL (Read, Execute, Print, Loop)'. The status bar at the bottom shows 'Ln 1, Col 1 Spaces: 4 UTF-8 LF Python 3.11.1 64-bit'.

A
Explorer

B ← *Edit a module*

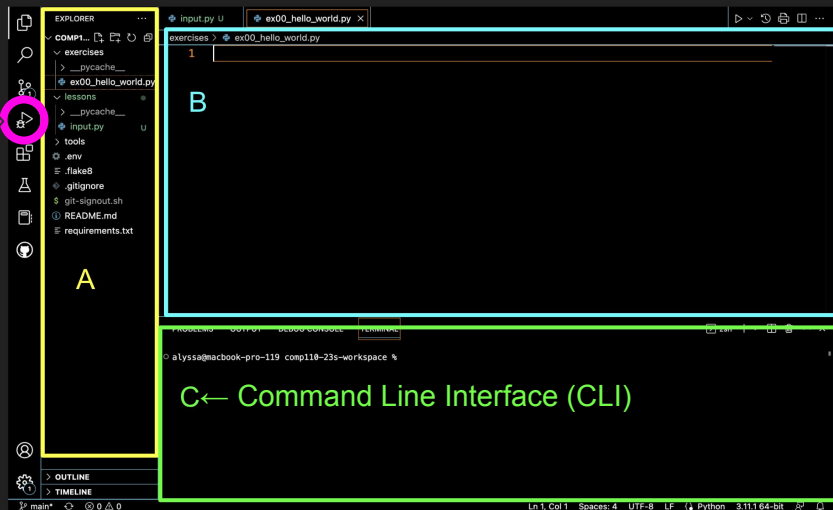
Editor

C ← Runs your Command Line Interface (CLI)

- *Run a module*
- *REPL (Read, Execute, Print, Loop)*

main* 0 0 0 Ln 1, Col 1 Spaces: 4 UTF-8 LF Python 3.11.1 64-bit

Ways to run code



Use Trailhead:

- Launch with the debug button
- “Starting Trailhead server at <http://localhost:1110>”

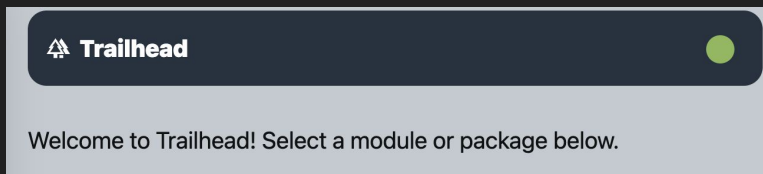
Interactive (like a conversation with your computer):

REPL: Read Execute Print Loop

- To initialize the REPL in your terminal, type:
 - `python`
- `>>>` means you're in the REPL

To run a module (execute a python (.py) file) from your terminal, type:

- `python -m my_file_name`



Objects and Built-In Types

An **object** is *typed* unit of data in memory.

The object's **type** classifies it to help the computer know how it should be interpreted and represented.

Programming languages offer many built-in data types for you to work with, typically including:

- numerical
 - integers
 - decimal numbers
 - complex numbers
- textual
- logical
- collections of many objects
 - sequences
 - sets
 - dictionaries

Numerical Built-In Types

- Integers

- `int`
- Zero, *or* non-zero digit followed by zero or more integers
- 100 is an int, but 0100 is not
- 3 is, but 3.08 is not
- -2000 is, but -2000.1 is not

- Floats (decimals)

- `float`
- Examples: 3.02, 4008.0, -16.99999
- Not the *only* way to represent decimal numbers, but a very precise way

Boolean Built-In Type

- `bool`
- Evaluates to `True` or `False`
- Important: these should *always* have a capital T or F!
 - `True` is a boolean value
 - `TRUE` and `true` are not
 - `False` is a boolean value
 - `FALSE` and `false` are not

Textual Built-In Type

- Strings

- `str`
- A sequence (or *string*) of characters
- Can be denoted using “ ”
- Examples:
 - A word: “hello”
 - A phrase: “Hope we get some snow!”
 - A single character: “A”, “ “, “🎉”
 - A number *in quotes*: “23”, “110”, “12.5”
 - An empty string: “”

Indexing

- **Subscription** syntax uses square brackets and allows you to access an item in a sequence
- **Index numbering starts from 0 (in Python)**

Example:

The string, “**happy**”

Indexing

- **Subscription** syntax uses square brackets and allows you to access an item in a sequence
- **Index numbering starts from 0 (in Python)**

Example:

Characters: **h a p p y**

The string, **"happy"**

Indices: **0 1 2 3 4**

"happy" [0] would give us what letter?



In English: "happy at index 0"

Docstrings

- A string written at the top of every file to describe its purpose.
- Denoted with three quotations `""" """`

Check an Object's Type

- `type()`
 - `type(3)`
 - `type("hi")`
 - `type(True)`

Change (Cast) an Object's Type

- `float()`
- `str()`
- `int()`

Homework!

- Respond to **Lesson 01 (LS01)** and **Lesson 02 (LS02)** Gradescope Questions → due **today** at 11:59pm
 - **LS01:** VS Code, Terminal, + Running a Program
 - **LS02:** Objects and Data Types
- Course Setup + **EX00** (due Wed, Jan 15th at 11:59pm)
 - Come to the open house in SN008 for help!
 - 11am-5pm today (unless University closes early)
 - 1-5pm Sunday
 - Normal office hours begin on Monday (11am-5pm)