



CL13 – Variables

Today will be a pencil and paper/tablet kind of day!

Announcements

Re: Quiz 01:

- Great job! We'll have these returned back to you soon.
- Once grades are published, [visit Office Hours and Tutoring](#) to go over quiz questions you missed – we *want* to help you!

LS 09: Variables due tonight at 11:59pm

EX02: Chardle

- Uses concepts from last unit and today
- A first step in your implementation of Wordle!
- Posted by Feb 19

Warm-Up: Discuss these questions with a neighbor, then diagram how you believe this works:

```
1  def f(x: int) -> int:
2  |     y: int
3  |     y = x * 2
4  |     return y
5
6
7  print(f(3))
```

Questions to discuss with a neighbor:

What does line 2 remind you of?

What does line 3 remind you of?

```
1 def f(x: int) -> int:
2     y: int
3     y = x * 2
4     return y
5
6
7 print(f(3))
```

```
1 def pizza_price(size: int) -> float:
2     """Calculate the price of a pizza."""
3     price: float = 10.0
4
5     if size >= 16:
6         price = 20.0
7
8     return price
9
10
11 print(pizza_price(size=16))
```

Key Variable Terminology

Variable Declaration / Definition - Associates a name/identifier with a data type, and a space in the current frame

`<name>: <type>`

Examples:

`students: int`

`message: str`

Key Variable Terminology

Variable Declaration / Definition

`<name>: <type>`

- Associates a name/identifier with a data type, and a space in the current frame

Variable Assignment

`students = 300`

- Binds a new value to a variable name in memory

Key Variable Terminology

Variable Declaration / Definition

`<name>: <type>`

- Associates a name/identifier with a data type, and a space in the current frame

Variable Assignment

`students = 300`

- Binds a new value to a variable name in memory

Variable Initialization

- First time a variable is assigned

Key Variable Terminology

Variable Declaration / Definition

`<name>: <type>`

- Associates a name/identifier with a data type, and a space in the current frame

Variable Assignment

`students = 300`

- Binds a new value to a variable name in memory

Variable Initialization

- First time a variable is assigned

Variable Access

- “Reading” or using a variable name in an expression

Left-hand vs. Right-hand Side of Assignment

Each side of the assignment operator (=) plays a distinct role in variable assignment!

Identify key concepts, then trace the program in a diagram!

Identify: Declaration, Initialization vs. Assignment, Access

```
1  def pizza_price(size: int, toppings: int) -> float:
2      """Calculate the price of a pizza with toppings."""
3      price: float = 10.0
4
5      if size >= 16:
6          price = 20.0
7
8      price = price + toppings * 0.75
9
10     return price
11
12
13  print(pizza_price(size=14, toppings=2))
```

```
1 def pizza_price(size: int, toppings: int) -> float:
2     """Calculate the price of a pizza with toppings."""
3     price: float = 10.0
4
5     if size >= 16:
6         price = 20.0
7
8     price = price + toppings * 0.75
9
10    return price
11
12
13 print(pizza_price(size=14, toppings=2))
```

Common Variable Errors

`UnboundLocalError` – Occurs when attempting to access a variable that is declared in a function but not yet initialized

`NameError` – Occurs when attempting to access a variable that has not been declared. Commonly from typos or renaming a variable and not updating all accesses

Why variables? One reason: to store the results of function calls for later use!

```
1  def pizza_price(size: int, toppings: int) -> float:
2      """Calculate price of pizza with toppings."""
3      price: float = 10.0
4
5      if size >= 16:
6          price = 20.0
7
8      price = price + toppings * 0.75
9
10     return price
11
12
13     total_price: float = pizza_price(size=14, toppings=2)
14     print(total_price)
```

```
1 def pizza_price(size: int, toppings: int) -> float:
2     """Calculate price of pizza with toppings."""
3     price: float = 10.0
4
5     if size >= 16:
6         price = 20.0
7
8     price = price + toppings * 0.75
9
10    return price
11
12
13 total_price: float = pizza_price(size=14, toppings=2)
14 print(total_price)
```