Python Variables and Types

What is a variable?

In Python:

```python
age = 11
height = 1.45
student = "Chelsea"
```

Here:

- `age`, `height`, `student` are **variable names**
- `11`, `1.45`, `"Chelsea"` are the **values**

Create variables in Python

We use the **equals sign** `=` to store a value in a variable.

```python
age = 11                # age is variable name, 11 is value
height = 1.45           # height is variable name, 1.45 is value
student = "Chelsea"     # student is variable name, "Chelsea" is value
```

Type 1: int (integer)

`int` means **integer**:

- Whole numbers
- No decimal point

Examples of **int**:

```
age = 11
year = 2026
temperature_today = -2
```

Type 2: float (decimal number)

`float` means a number with a decimal point (like 3.0, 2.5).

Examples of `float`:

```
pi = 3.142
height = 1.45
temperature = 26.5
```

Type 3: str (string / text)

`str` means a string or a piece of text.

Examples:

```
name = "Chelsea"
favorite_food = "ice cream"
greeting = "Hello, Python!"
```

Remember: Put text inside **quotes**: `" "` or `' '`

Print a message

To show something on the screen, we use `print`.

```python
print("Hello, Python!")
print("Welcome to variables!")
print("This is fun!")
```

Print a message

To show something on the screen, we use `print`.

```
print("Hello, Python!")
print("Welcome to variables!")
print("This is fun!")
```

Output:

```
Hello, Python!
Welcome to variables!
This is fun!
```

- Python prints each message on its own line.
- The text must be in quotes to be a string.

Print variables

We can print the **value** stored in a variable.

```
age = 11
student = "Chelsea"

print(age)
print(student)
```

Print variables

We can print the **value** stored in a variable.

```
age = 11
student = "Chelsea"

print(age)
print(student)
```

Output:

```
11
Chelsea
```

Print text + variables together

- We can print **strings** and **variables** in one `print`.
- They are separated by a comma `,`

```
student = "Chelsea"
age = 11

print("Student:", student)
print("Age:", age)
```

Print text + variables together

- We can print **strings** and **variables** in one `print`.
- They are separated by a comma `,`

```
student = "Chelsea"
age = 11

print("Student:", student)
print("Age:", age)
```

Output:

```
Student: Chelsea
Age: 11
```

`print` puts a **space** between items automatically.

Print multiple values on one line

You can print many items at once:

```
x = 5
y = 7

print("x =", x, "y =", y)
```

Print multiple values on one line

You can print many items at once:

```python
x = 5
y = 7

print("x =", x, "y =", y)
```

Output:

```
x = 5 y = 7
```

- All in **one line**.
- `print` puts a **space** between items automatically.

Very important: quotes vs no quotes

Python code:

```python
student = "Chelsea"
print(student)
print("student")
```

Very important: quotes vs no quotes

Python code:

```
student = "Chelsea"
print(student)
print("student")
```

Output:

```
Chelsea
student
```

Very important: quotes vs no quotes

Python code:

```
student = "Chelsea"
print(student)
print("student")
```

Output:

```
Chelsea
student
```

What is the difference?

- `student` (no quotes) means "use the **value stored in** the variable".
- `"student"` (with quotes) is a **string** that literally says: `student`.

Check the type of a variable

Python can tell you the **type** of a variable using `type(...)`.

```python
age = 11
height = 1.45
student = "Chelsea"

print(type(age))
print(type(height))
print(type(student))
```

Check the type of a variable

Python can tell you the **type** of a variable using `type(...)`.

```
age = 11
height = 1.45
student = "Chelsea"

print(type(age))
print(type(height))
print(type(student))
```

Output:

```
<class 'int'>
<class 'float'>
<class 'str'>
```

Q1 🔢 What are the types?

Python code:

```python
a = 10
b = 2.5
c = "Hello"
d = "123"
```

Fill in the blanks:

1. `a` is ______
2. `b` is ______
3. `c` is ______
4. `d` is ______

Q1 🔢 What are the types?

Python code:

```
a = 10
b = 2.5
c = "Hello"
d = "123"
```

Answers:

1. `a` is `int` (integer)
2. `b` is `float`
3. `c` is `str` (string)
4. `d` is `str` (string), even though it looks like a number

Q2 🖨️ What does this print?

Python code:

```python
name = "Alex"
score = 95

print("Player:", name)
print("Score:", score)
```

Q2 🖨️ What does this print?

Python code:

```python
name = "Alex"
score = 95

print("Player:", name)
print("Score:", score)
```

Output:

```
Player: Alex
Score: 95
```

Q3 🧪 Guess the types with `type`

What will this program print?

```
x = 3
y = 3.0
z = "3"

print(type(x))
print(type(y))
print(type(z))
```

Q3 🧪 Guess the types with `type`

What will this program print?

```
x = 3
y = 3.0
z = "3"

print(type(x))
print(type(y))
print(type(z))
```

Output:

```
<class 'int'>
<class 'float'>
<class 'str'>
```

Summary

- Use `=` to store a value in a variable.
- Basic types:
  - `int` → whole numbers
  - `float` → decimal numbers
  - `str` → text (strings)
- `print(variable)` shows variable value.
- `type(variable)` tells you the **type** of that variable.

ALGORITHM
WITH CHELSEA