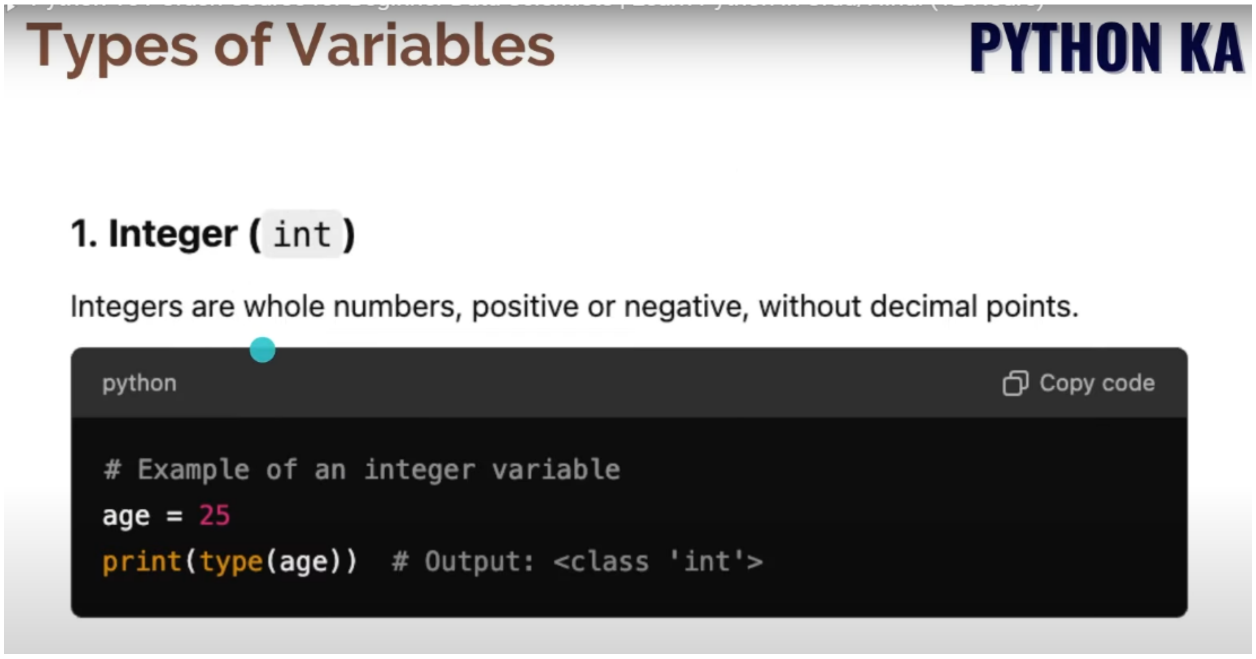


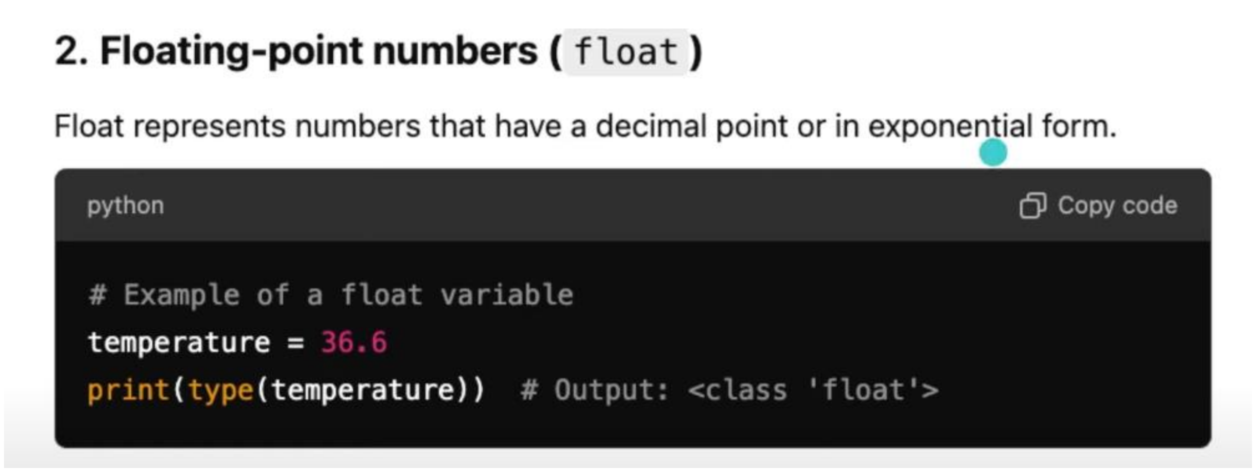
Types of Variables

PYTHON KA

1. Integer (`int`)

Integers are whole numbers, positive or negative, without decimal points.

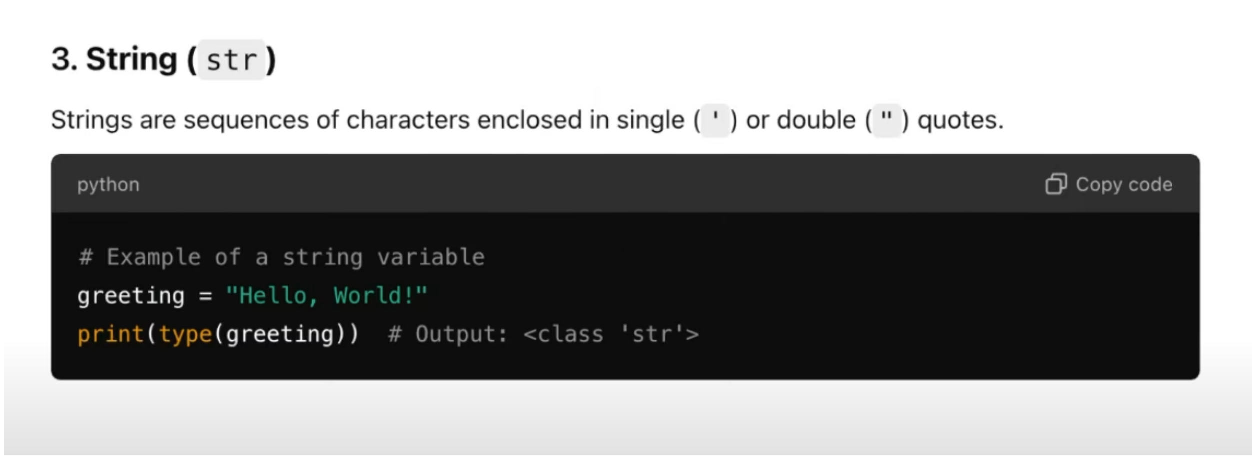
```python
# Example of an integer variable
age = 25
print(type(age))  # Output: <class 'int'>
```



2. Floating-point numbers (`float`)

Float represents numbers that have a decimal point or in exponential form.

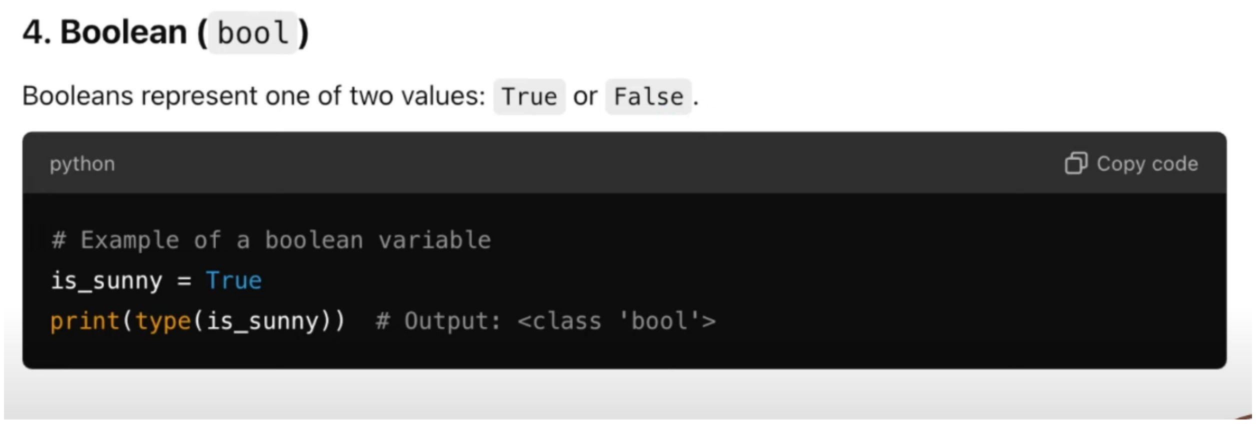
```python
# Example of a float variable
temperature = 36.6
print(type(temperature))  # Output: <class 'float'>
```



3. String (`str`)

Strings are sequences of characters enclosed in single (`'`) or double (`"`) quotes.

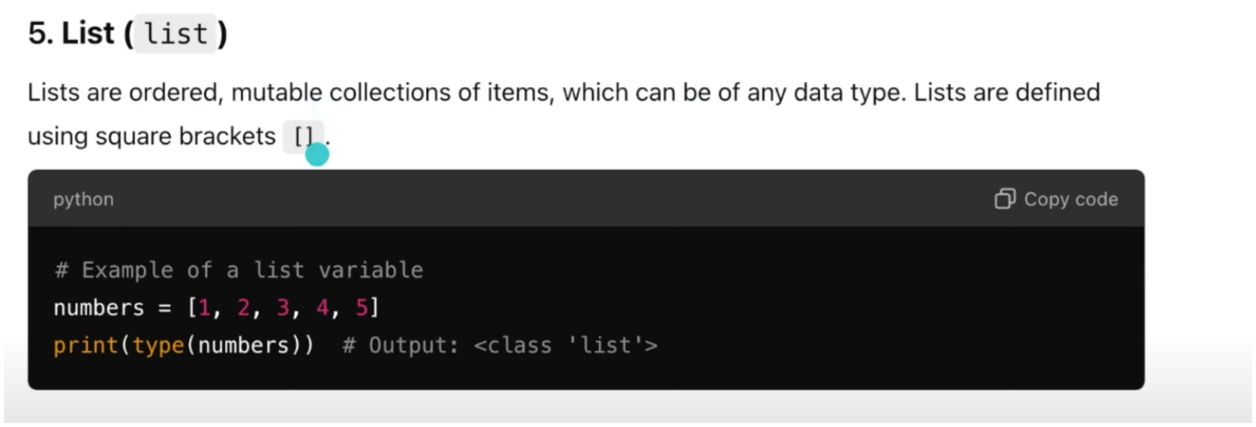
```python
# Example of a string variable
greeting = "Hello, World!"
print(type(greeting))  # Output: <class 'str'>
```



4. Boolean (`bool`)

Booleans represent one of two values: `True` or `False`.

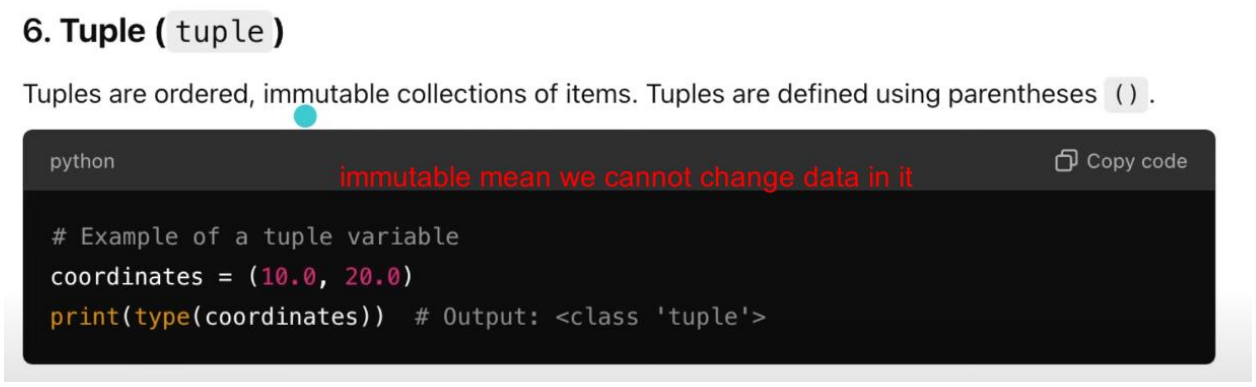
```python
# Example of a boolean variable
is_sunny = True
print(type(is_sunny))  # Output: <class 'bool'>
```



5. List (`list`)

Lists are ordered, mutable collections of items, which can be of any data type. Lists are defined using square brackets `[]`.

```python
# Example of a list variable
numbers = [1, 2, 3, 4, 5]
print(type(numbers))  # Output: <class 'list'>
```



6. Tuple (`tuple`)

Tuples are ordered, immutable collections of items. Tuples are defined using parentheses `()`.

immutable mean we cannot change data in it

```python
# Example of a tuple variable
coordinates = (10.0, 20.0)
print(type(coordinates))  # Output: <class 'tuple'>
```

7. Dictionary (`dict`)

Dictionaries are unordered collections of key-value pairs, defined using curly braces `{}`.

python Copy code

```python
# Example of a dictionary variable
person = {"name": "Alice", "age": 30}
print(type(person))  # Output: <class 'dict'>
```

8. Set (`set`)

Sets are unordered collections of unique items, defined using curly braces `{}` or the `set()` function.

python Copy code

```python
# Example of a set variable
fruits = {"apple", "banana", "orange"}
print(type(fruits))  # Output: <class 'set'>
```

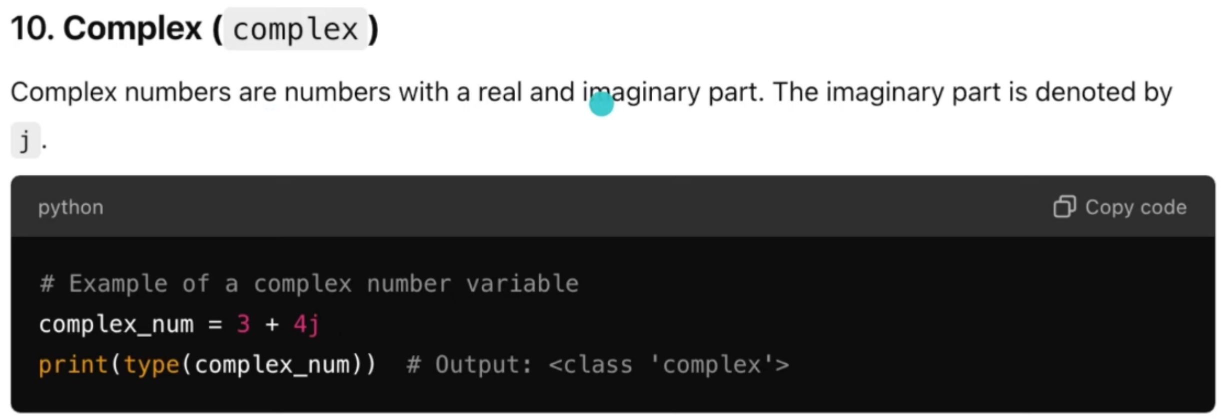
9. None (`NoneType`)

`None` represents the absence of a value and is an object of its own datatype.

python Copy code

```python
# Example of NoneType variable
result = None
print(type(result))  # Output: <class 'NoneType'>
```

Complex is used very rare J is used for complex keyword we cannot use other keyword accept the j

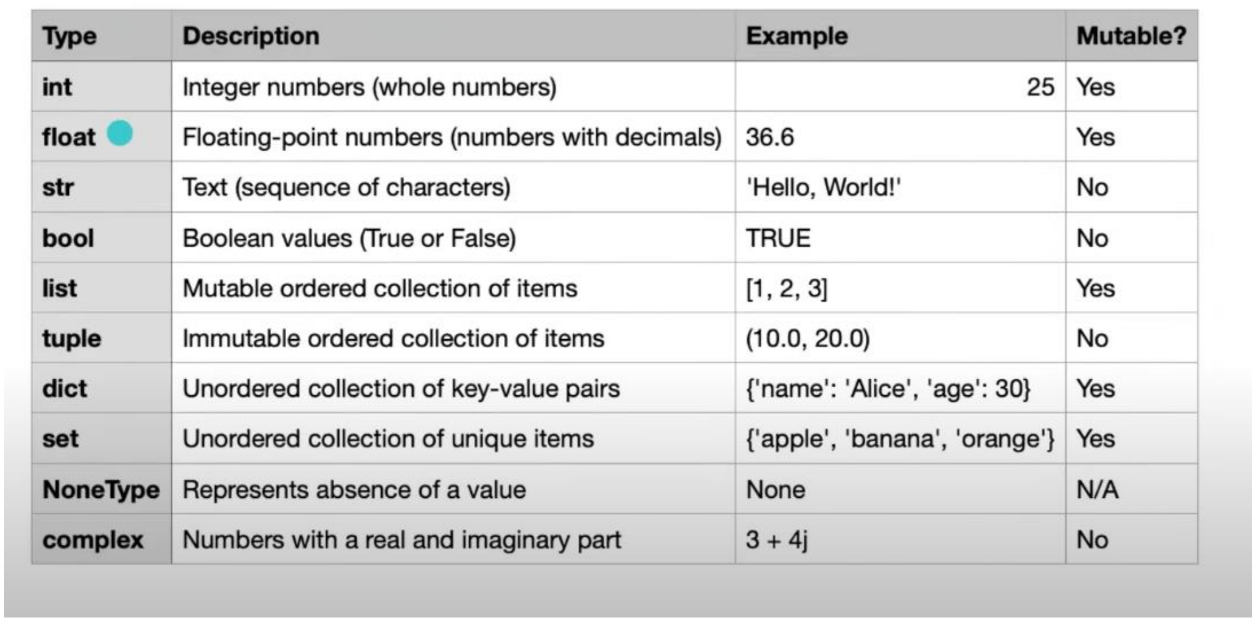


10. Complex (`complex`)

Complex numbers are numbers with a real and imaginary part. The imaginary part is denoted by `j`.

```python
# Example of a complex number variable
complex_num = 3 + 4j
print(type(complex_num))  # Output: <class 'complex'>
```

Mutable mean we can change



Type	Description	Example	Mutable?
int	Integer numbers (whole numbers)	25	Yes
float	Floating-point numbers (numbers with decimals)	36.6	Yes
str	Text (sequence of characters)	'Hello, World!'	No
bool	Boolean values (True or False)	TRUE	No
list	Mutable ordered collection of items	[1, 2, 3]	Yes
tuple	Immutable ordered collection of items	(10.0, 20.0)	No
dict	Unordered collection of key-value pairs	{'name': 'Alice', 'age': 30}	Yes
set	Unordered collection of unique items	{'apple', 'banana', 'orange'}	Yes
NoneType	Represents absence of a value	None	N/A
complex	Numbers with a real and imaginary part	3 + 4j	No