Python Basics

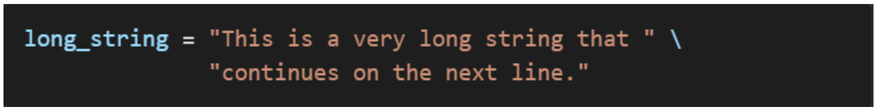
Types of Errors

Python encounters various types of errors during execution, including:

- **Syntax Errors** ⟶ Occur when the Python interpreter finds an incorrectly written statement.
- **Name Errors** ⟶ Occur when a variable or function name is not defined.
- **Type Errors** ⟶ Happen when an operation is applied to an object of an inappropriate type.
- **Index Errors** ⟶ Raised when trying to access an index that is out of range in a sequence.
- **Key Errors** ⟶ Raised when a dictionary key is not found.
- **Attribute Errors** ⟶ Occur when an invalid attribute is accessed on an object.

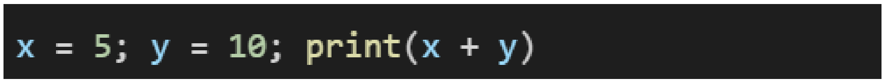
Basic Syntax Rules

- Case Sensitivity - Python is case-sensitive, meaning Variable and variable are different identifiers.
- Indentation - Python uses indentation (whitespace) to define blocks of code. Incorrect indentat leads to errors.
- Line Continuation - Use a backslash (\) to continue a statement on the next line:



```
long_string = "This is a very long string that " \
              "continues on the next line."
```

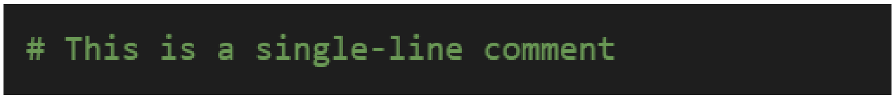
- Multiple Statements on a Single Line - Use a semicolon (;) to separate multiple statements:



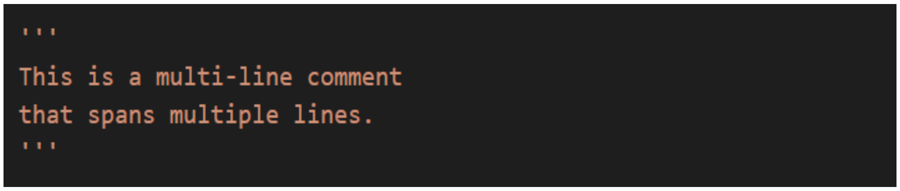
```
x = 5; y = 10; print(x + y)
```

Comments in Python

- Single Line Comment
- Multi- Line Comment



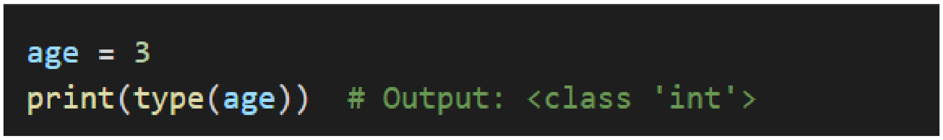
```
# This is a single-line comment
```



```
'''
This is a multi-line comment
that spans multiple lines.
'''
```

Variable Assignment

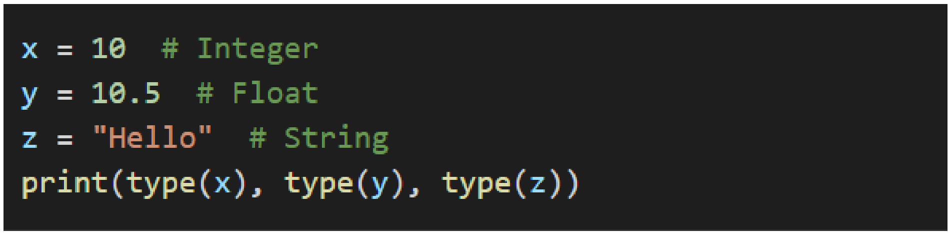
Python determines the type of a variable dynamically:



```
age = 3
print(type(age))  # Output: <class 'int'>
```

Type Inference

Python decides the data type during runtime, allowing dynamic type changes:



```
x = 10  # Integer
y = 10.5  # Float
z = "Hello"  # String
print(type(x), type(y), type(z))
```