

Variables in Python

In Python, a variable is a named storage location that holds a value. It's like a container that can store different types of data, such as numbers, text, or even more complex objects.

Creating Variables

To create a variable in Python, you simply assign a value to it using the assignment operator (=). The syntax is as follows:

```
In [2]: x = 10
        name = "Alice"
        is_valid = True
```

Here, x is a variable that refers to the integer 10, name is a variable that refers to the string "Alice", and is_valid is a variable that refers to the boolean value True.

Variable Types

Python is a dynamically typed language, which means you don't need to explicitly declare the type of a variable before using it. The type of a variable is determined by the value it holds. Python has several built-in data types, including:

Numbers: Integers (e.g., 10), floating-point numbers (e.g., 3.14), and complex numbers.

Strings: Sequences of characters (e.g., "Hello").

Lists: Ordered collections of elements (e.g., [1, 2, 3]).

Tuples: Immutable ordered collections of elements (e.g., (1, 2, 3)).

Dictionaries: Unordered collections of key-value pairs (e.g., {'name': 'Alice', 'age': 30}).

Booleans: True or False values.

example

```
In [5]: num = 10          # Integer
        pi = 3.14        # Floating-point number
        message = "Hello, world!" # String
```

Rules to create a variables

A variable name cannot start with a number or any special character except underscore '_'.

A variable name must start with a letter or the underscore character

A variable name can only contain alpha-numeric characters and underscores(A-Z,0-9,_)

A variable name cannot be any of the python keywords

Python variable names are case-sensitive(age, Age, AGE are three different variables)

Valid variables

Start with a letter (a-z or A-Z) or an underscore (_): This ensures that the variable name is distinguishable from other language elements.

Contain only letters, numbers, and underscores: Avoid using special characters or spaces.

Be case-sensitive: myVariable and myvariable are considered different variables.

Not be a Python keyword: Avoid using reserved words like if, else, for, while, etc.

example

```
In [7]: x = 10
        my_variable = "Hello"
        _private_var = True
        age_25 = 25
```

Invalid Variables

Start with a number: 123variable is invalid.

Contain spaces or special characters: my variable and my#variable are invalid.

Be a Python keyword: if = 5 is invalid.

example

```
In [8]: 123variable = 10
        my variable = "Hello"
        my#variable = True
        if = 5

Cell In[8], line 1
    123variable = 10
    ^
SyntaxError: invalid decimal literal
```

20 valid variables

```
In [9]: # Valid variables:

# Starting with a letter
name = "Alice"
age = 25
height = 1.75

# Starting with an underscore
_private_var = True

# Containing letters, numbers, and underscores
my_variable_1 = 10
variable_with_number = "Hello"

# Case-sensitive
myVariable = "Value"
myvariable = "Another Value"

# Using conventions (snake_case and camelCase)
snake_case_variable = 123
camelCaseVariable = "Example"

# Using meaningful names
customer_name = "John Doe"
product_price = 9.99

# Using multiple words separated by underscores
first_name = "Alice"
last_name = "Smith"

# Using abbreviations
temp = 25
url = "https://example.com"
```

20 invalid variables

```
In [10]: # Invalid variables:

# Starting with a number
123variable = 10

# Containing spaces
my variable = "Hello"

# Containing special characters
my#variable = True

# Using Python keywords
if = 5
for = 10

# Using reserved words
and = True
or = False

# Using non-ASCII characters (without proper encoding)
é = "é"

# Using overly long names
extremely_long_variable_name_that_is_difficult_to_read = 1

# Using single-letter names (except for common temporary variables)
a = 1
b = 2

# Using confusingly similar names
variable1 = 10
variable2 = 20

# Using names that are hard to pronounce or remember
x_y_z = 123
abcde = "Example"

Cell In[10], line 4
    123variable = 10
    ^
SyntaxError: invalid decimal literal

In [ ]:
```