Data Types, Operators, and Variables

- Data Types

- Relational Operators

- Assignment Operators

- Variables

- Correct / Incorrect Examples

- References

Data Types

In programming we work with different types of data. Here we are going to talk about four types of data types.

Data Type	Name	Examples		
`str`	`String`	"Dave"	"230"	"What is your name? "
`int`	`Integer`	43	200	-320390920283
`float`	`Floating-point`	25.50	-235.75	2.15443454545345

Relational Operators

Operator	Name	Description
`==`	`Equal to`	Returns True if both operands are equal.
`!=`	`NOT Equal`	Returns True if the left and right operands are not equal.
`>`	`Greater than`	Returns True if the left operand is greater than the right operand.
`<`	`Less than`	Returns True if the left operand is less than the right operand.
`>=`	`Greater than OR equal to`	Returns True if the left operand is greater than or equal to the right operand.
`<=`	`Less than OR equal to`	Returns True if the left operand is less than or equal to the right operand.

Assignment Operators

Operator	Name	Description
`=`	`Num = 5`	Creates a variable and assigns the value on right to the variable on the left.
`+=`	`Count += 1`	Same as: Count = Count + 1
`-=`	`Count -= 1`	Same as: Count = Count - 1
`*=`	`Tax *= .05`	Same as: Tax = Tax * .05
`/=`	`Greater than OR equal to`	Returns True if the left operand is greater than or equal to the right operand.
`<=`	`Less than OR equal to`	Returns True if the left operand is less than or equal to the right operand.

Variables

Variables are a temporary storage area for data.

Rules for naming Variables

1. Must start with a letter OR underscore

2. Cannot start with a number

3. Cannot contain spaces

4. Can only contain alpha-numeric characters and underscore (A-z, 0-9, _)

4. Variable names are case sensitive. Dave, DAve, DAVe, and DAVE are all different variable names.

5. Cannot be a Python reserved keyword (i.e., and, if False, True, etc..)

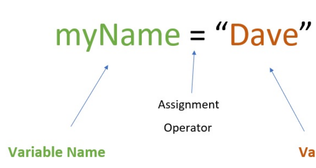
Recommendations for naming variables in Python

1. Start with a lowercase letter.

2. Use Camel case (i.e., myNumber, intCounter)

3. Use meaningful names based on the data they hold. Names that are easy to remember

4. Don't use built-in names like type, print, def, etc..



Correct Examples

The below examples are all string variables.

In [1]:
```
#Legal variable names:
myvar = "Dave"
my_var = "Dave"
_my_var = "Dave"
myVar = "Dave"
MYVAR = "Dave"
myvar2 = "Dave"
```

The below examples are all number variables.

In [2]:
```
#Legal numeric variable names:

num1 = 44
num2 = 3423423.234234
num3 = -32.55
num4 = 5j
```

Incorrect Examples

These examples show exemption errors because of the variable name. The string "Dave" is correct throughout these examples.

In [3]:
```
#Illegal variable names:

2myvar = "Dave"
```

```
  File "<ipython-input-3-43e8a6b13cf0>", line 3
    2myvar = "Dave"
    ^
SyntaxError: invalid syntax
```

In [4]:
```
#Illegal variable names:

my-var = "Dave"
```

```
  File "<ipython-input-4-6de069e3f210>", line 3
    my-var = "Dave"
       ^
SyntaxError: can't assign to operator
```

In [5]:
```
#Illegal variable names:

my var = "Dave"
```

```
  File "<ipython-input-5-39775f7c6a16>", line 3
    my var = "Dave"
       ^
SyntaxError: invalid syntax
```

Resources

Sources for deeper learning:

1. W3Schools.com Python Variables

1. Python Variables: - Python Tutorial for Beginners with Examples.

Sources used in this document:

Python Booleans from W3Schools.com

Python Variables from W3Schools.com

Variables in Python, from RealPython.com