

Data types in python

Python is a dynamically typed language, which means you don't need to declare the data type of a variable when you create it; Python will determine the data type dynamically at runtime. However, Python has several built-in data types that you can use to store and manipulate different kinds of data. Here are some of the most common data types in Python:

- Numeric - Numeric variables take values which are numbers like 9, 3.14, 0, Inf
- String - 'hello','hello'
- Boolean - True, False
- Datetime - 2023-08-05 08:21:04

✓ Integers and Floats

```
a = 100.0
```

```
print(type(a))

<class 'float'>
```

```
aB=10.12
Ab=5
print(type(Ab))

<class 'int'>
```

```
a = False
print(type(a))

<class 'bool'>
```

```
print(type(False))

<class 'bool'>
```

```
a = "hello"
print(type(a))

<class 'str'>
```

```
a = 10
print(a)
print(type(a))
print(type('hello'))

10
<class 'int'>
<class 'str'>
```

```
a = 'True'
print(type(a))

<class 'str'>
```

✓ Perform Arithmetic oprations

```
print(10+10)

20
```

```
print(10+12)
```

```
22
```

```
# Addition opration  
print(2+44)
```

```
46
```

```
# Subtraction opration  
10-10
```

```
0
```

```
# Multiplication opration  
3*3
```

```
9
```

```
a = 223.0  
print(type(a))  
a = int(a)  
print(a)  
  
<class 'float'>  
223
```

```
print(int(10.99))
```

```
10
```

```
10/5
```

```
2.0
```

```
# Division opration  
a = 10  
b = 5  
c = a/b  
print(int(c))
```

```
2
```

```
# Modulus opration  
9%3
```

```
0
```

```
13/2
```

```
6.5
```

```
# Floor Division opration  
13//2
```

```
6
```

```
3*3
```

```
9
```

```
# Exponentiation opration  
2**3
```

```
8
```

```
# Python follow BODMAS
```

```
2+3+5-3*2
```

```
4
```

```
2+ 10* (10+3)
```

```
132
```

```
a = 112000
```

```
print(int(a))
```

```
900000
```

```
# large numbers
```

```
(int(112E3))
```

```
112000
```

✓ Let's Know a bout Variable?

✓ VARIABLES are entities which help us store information and retrieve it later.

```
# Variables
```

```
x=20
```

```
print(x)
```

```
20
```

```
y=6.9
```

```
print(type(y))
```

```
<class 'float'>
```

```
z = x+y
```

```
print(int(z))
```

```
26
```

✓ Basic Arithmetic operations performed on variables

```
# Addition operation
```

```
z = x+y
```

```
print(z)
```

```
print(int(z))
```

```
print(x)
```

```
print(y)
```

```
26.9
```

```
26
```

```
20
```

```
6.9
```

```
# Subtraction operation
```

```
z = y-x
```

```
print(z)
```

```
-13.1
```

```
x =10
```

```
print(x)
```

```
10
```

```
print(x)

10

x = 5
print(x)

5

print(z)

-13.1

print(z)

# Check about data type of y
print(y)
type(y)

6.9
float

# Multiplication operation
z = x/2

print(z)
z = int(z)
type((z))

2.5
int

z

2

# Division operation
z = x/y

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

0.7246376811594203
5
6.9
float

20/3

6.666666666666667

x = 20
y = 3

# Floor division operation
z= x//y
print(z)

6

x

20

# Modulo operato
y=5
x=3
```

```
z = y%x
7
```

```
2
```

```
3*3*3*3*3
243
```

```
# Using powers and exponents
```

```
z = x**y
print(z)

243
```

```
# B - Brackets, O - Order of powers or roots, D - Division, M - Multiplication A - Addition, and S - !
operation =(x+y)/y + (y-x)*x + x - y
```

```
print(operation)
type(operation)

5.6
float
```

✓ Rules for naming a variable in Python

- Variables names must start with like *a* , *A*
- toppers1, topperS1, toppers_academy are some valid variable
- variables are case sensitive

```
Herry_data = 10
print(Herry_data)
```

```
10
```

```
1_a_ = 10
print(1_a_)
```

```
File "<ipython-input-103-4372c58d8123>", line 1
```

```
1_a_ = 10
^
```

```
SyntaxError: invalid decimal literal
```

Next steps:

[Fix error](#)

```
_alpha1 = 4
_alpha2 = 10
_alpha3 = 22
print(_alpha3)
```

```
22
```

```
a2 = 3
print(a2)
```

```
3
```

```
a = 10
A= 20
print(a)
print(A)
```

```
10
20
```

```
the_price_of_car = 10000

print(the_price_of_car)

10000
```

```
_abc_1 = 2
_abc_1

2
```

```
a = 10
```

```
type = 5
print(type)

5
```

```
type(a)

int
```

```
true = 10
true

10
```

- Names can not contain any of these symbols:

```
@#%^&*~+:'",<>/?|\!
```

- we can not use python function as variable

✓ Boolean Variables

- True or False

✓ Comparison Operators

Equal to ==

Not equal to !=

Greater than >

Less than <

Greater than or equal to >=

Less than or equal to <=

```
# Set object to be a boolean
boolean_var = False
type(boolean_var)

bool

#Show
boolean_var

False
```

✓ Equal

```
a = 10  
b = 1
```

```
a==b  
  
False
```

```
2 == 5  
  
False
```

```
2==0  
  
False
```

✓ Not equal

```
x = 2  
y = 3  
  
x == y  
  
False
```

```
2!=2  
  
False
```

```
2==2  
  
True
```

✓ Greater than

```
a=3  
b=7  
b > a  
  
True
```

```
a == 3  
  
True
```

```
b > 4  
  
True
```

✓ Less than

```
10 < 45  
  
True
```

```
4 < 2  
  
False
```

✓ Greater than or equal to

```
2 >=2  
  
True
```

```
4 >= 4
```

```
True
```

▼ Less than or equal to

```
3 <= 0
```

```
False
```

```
1 <= 2
```

```
True
```

```
a = 10
```

```
b = 20
```

```
a<b
```

```
True
```

Start coding or [generate](#) with AI.