

// NUMBER TYPES

```
age = 25          # int
height = 1.80    # float
type(age)        # <class "int">
type(height)     # <class "float">
```

No quotes = number. Python assigns type automatically.

// int vs float

```
10 / 2           # 5.0 - always float
10 // 2          # 5   - int
int(3.9)         # 3   - truncates
float(5)         # 5.0
5 + 2.0         # 7.0 - mixing = float
0.1 + 0.2       # 0.300000...4
```

Float precision - not a bug, how it works

// ROUNDING

```
round(3.14159, 2) # 3.14
round(3.14159, 4) # 3.1416
round(3.5)        # 4
round(2.5)        # 2 - banker's rounding
```

Rounds to nearest even when exactly halfway

// INPUT WITH NUMBERS

```
input() always returns a string
age = int(input("Age: "))
height = float(input("Height: "))
f"Age: {age} | Height: {height}"
```

// ARITHMETIC OPERATORS

```
10 + 3  # 13 - addition
10 - 3  # 7  - subtraction
10 * 3  # 30 - multiplication
10 / 3  # 3.333... - always float
10 // 3 # 3  - floor division (int)
10 % 3  # 1  - modulo (remainder)
10 ** 3 # 1000 - exponentiation
```

// CONVERSION

```
int(3.9)    # 3   - truncates
int("25")   # 25  - string to int
float(5)    # 5.0
float("3.14") # 3.14
str(25)     # "25" - to string
input() always returns string - convert before math
```

// MATH MODULE

```
import math
math.pi      # 3.14159...
math.sqrt(16) # 4.0
math.pow(2, 8) # 256.0
math.floor(3.9) # 3
math.ceil(3.1) # 4
import goes at the top of your file - always
```

// COMMON MISTAKES

```
! / always returns float - use // for int
! int() truncates - use round() for rounding
! Float precision quirk - 0.1+0.2 != 0.3
! input() is string - convert before math
! Division by zero -> ZeroDivisionError
! Precedence applies - use parentheses
```