

# CDS 230

# Modeling and Simulation I

## Module 2

### Variables and Basic Data Types in Python



Dr. Hamdi Kavak  
<http://www.hamdikavak.com>  
[hkavak@gmu.edu](mailto:hkavak@gmu.edu)



# Topics covered

1. Numbers
2. Mathematical functions
3. Variables



# Numbers (code)

## 1.1. Integers

```
[ ]: -1
[ ]: type(-1)
[ ]: 9
[ ]: 8234118
[ ]: 12312741717471289549
[ ]: 9999999999999999999999999999999
```

# is the comment character

## 1.2. Floats

```
[ ]: 1.0
[ ]: type(1.0)
[ ]: 4/3 # a real number with infinite decimal points
[ ]: 3/5
[ ]: 4.12434351e-10 # scientific notation
[ ]: 2.1e5 # another scientific notation
```

## 1.3. Complex numbers

```
[ ]: 3+5j
[ ]: type(3+5j)
[ ]: complex(3)
[ ]: complex(0,5)
```

# Basic arithmetic

- + addition
- subtraction
- \* multiplication
- / decimal division
- // integer division (*rounded down*)
- % modulus
- \*\* power

# Basic arithmetic (code)

`+` addition, `-` subtraction, `*` multiplication

```
[ ]: 5 + 9
```

```
[ ]: 2.3 + 7.7
```

```
[ ]: 0.1 + 0.2
```

```
[ ]: 99-11
```

```
[ ]: 6 * 4
```

```
[ ]: 0.1 * 99
```

# Basic arithmetic (code)

`/` decimal division, `//` integer division, `%` modulus, `**` power

```
[ ]: 5.0 / 2
```

```
[ ]: 5 / 2
```

```
[ ]: 5 // 2
```

```
[ ]: 5.0 // 2
```

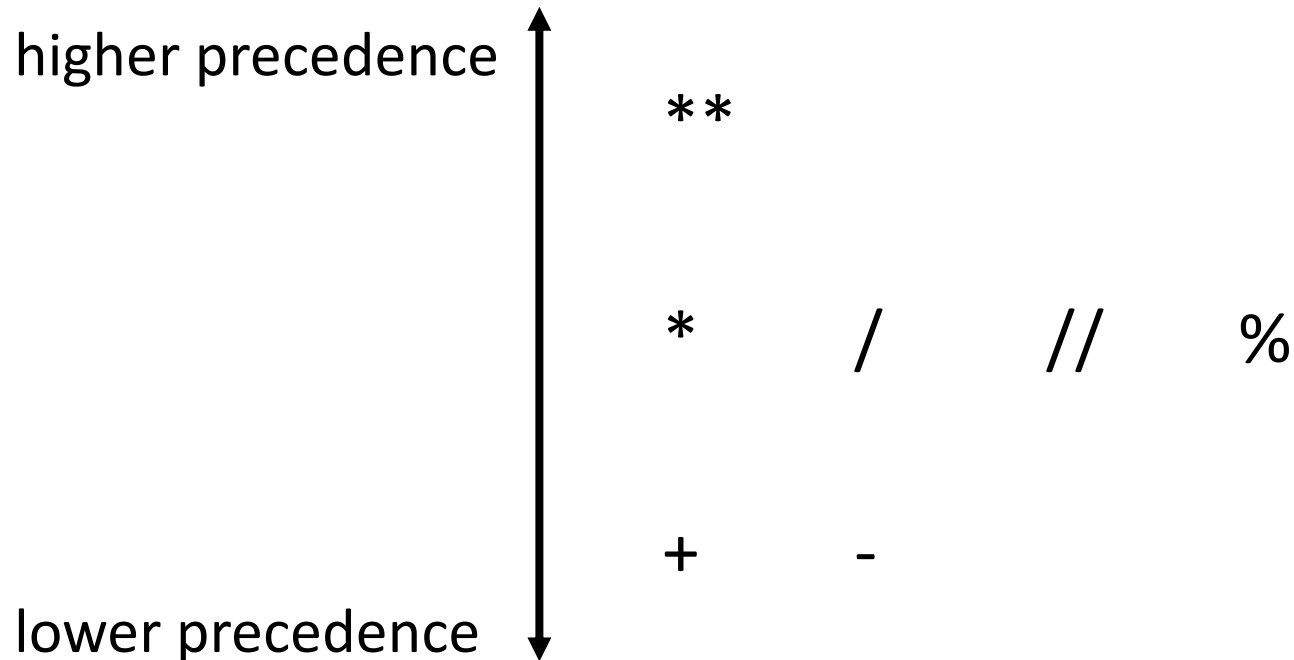
```
[ ]: 9 % 2
```

```
[ ]: 9.9 % 3
```

```
[ ]: 2 ** 3
```

```
[ ]: 5.5 ** 3.3
```

# Operator precedence



Parentheses  
overrides the  
precedence

$$5 + 3 * 8 = ?$$

$$(5 + 3) * 8 = ?$$

If multiple  
operators have the  
same precedence,  
code is executed  
from left to right



# Operator precedence (code)

```
[ ]: 5 + 3 * 8
```

```
[ ]: (5 + 3) * 8
```

```
[ ]: 2 + 10 / 2 * 5
```

```
[ ]: 2 + 32 / 2 ** 5
```

# Mathematical functions

- Have a **name**, take **value(s)** as **argument(s)**, and **return value(s)**.
- Two built-in math functions in Python: `abs` and `round`
- E.g.: `abs(x)` → will return a value

↑  
name

↑  
parameter

```
[ ]: abs(-123)
```

```
[ ]: abs(2)
```

```
[ ]: abs(3+4j)
```

# The `round()` function

- Can you guess the answers?

[ ]: `round(3.4)`

[ ]: `round(3.7)`

[ ]: `round(3.5)`

[ ]: `round(-3.5)`

[ ]: `round(-2.5)`

# Other mathematical functions

- Requires to import `math` module: `import math`

## Some general functions

```
math.ceil(x)
math.factorial(x)
math.floor(x)
math.exp(x)
math.log(x[, base])
math.log10(x)
math.pow(x, y)
math.sqrt(x)
```

## Some trigonometric functions

```
math.cos(x)
math.sin(x)
math.tan(x)
math.acos(x)
math.asin(x)
math.atan(x)
...
math.degrees(x)
math.radians(x)
```

## Some constants

```
math.pi
math.e
```

<https://docs.python.org/3/library/math.html>

# Mathematical functions (code)

```
[ ]: import math
```

```
[ ]: math.ceil(3.7)
```

```
[ ]: math.factorial(5)
```

```
[ ]: math.exp(2)
```

```
[ ]: math.log10(10)
```

```
[ ]: math.sqrt(100)
```

```
[ ]: math.cos(0)
```

```
[ ]: math.pi
```

```
[ ]: math.e
```

```
[ ]: math.sin(math.pi/2)
```

# Variables

- We can use variables to store values for re-use in other operations
- A variable is composed of two parts



- Python replaces the name with value if the variable is called later

```
year_now = 2020
year_born = 1978
age = year_now - year_born
```

# Variable naming

- Try to name meaningfully but you have to follow some rules
  - Case sensitivity
  - Only letters, underscore (\_) and digits are allowed
    - Cannot start with a digit
  - Do not use the built-in constant names and reserved keywords
    - 'False', 'None', 'True'
    - 'and', 'as', 'assert', 'async', 'await', 'break', 'class', 'continue', 'def', 'del', 'elif', 'else', 'except', 'finally', 'for', 'from', 'global', 'if', 'import', 'in', 'is', 'lambda', 'nonlocal', 'not', 'or', 'pass', 'raise', 'return', 'try', 'while', 'with', 'yield'

# Inspecting variable value (code)

There are two ways to inspect the value of a variable

```
year_now = 2020
year_born = 1978
age = year_now - year_born
```

## 1. Calling variable name in the last line of a cell

```
year_now = 2020
year_born = 1978
age = year_now - year_born
age
```

42

## 2. Using print function

```
print(age)
```

42

```
print('RIP Kobe (' ,age, ')')
```

RIP Kobe ( 42 )



# Sources

- *Learning Scientific Programming with Python* by Christian Hill.