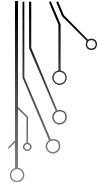# CLASS 05
FUNCTIONS & THE MATH MODULE

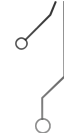COMP 130 – INTRODUCTION TO COMPUTING

DICKINSON COLLEGE

FUNCTIONS

- A *function* is a named sequence of statements that performs a computation.
  - a.k.a method / procedure / sub-routine
  - Some Python Built in Functions:
    - `type(3)`
    - `x=int('17')`
    - `print('Hello!')`

- A Function:
  - **Accepts** values as *arguments* (e.g. 3 or '17')
    - Enclosed in parenthesis.
  - *Returns a value* as a result (e.g. int or 17)
    - Can be used like any other value. (e.g. in expressions and assignments statements.)
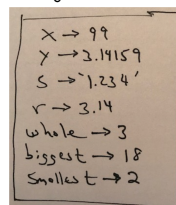   - E.g. `type(int('17'))`

PYTHON BUILT-IN FUNCTIONS

- Python provides many built-in functions
  - `x=int('99')`
  - `y=float('3.14159')`
  - `s=str(1.234)`

  - `r=round(y,2)`
  - `whole=round(y)`

  - `biggest=max(2, 7, 3, 18, 4)`
  - `smallest=min(5, 7, 2, 18, 4, 27)`

State Diagram


```
x → 99
y → 3.14159
s → '1.234'
r → 3.14
whole → 3
biggest → 18
smallest → 2
```

MODULES

- A Python *module* is a collection of related functions, variables and/or classes/objects.
  - Examples:
    - *math*: functions and variables for common mathematical operations.
    - *random*: provides functions for generating random numbers.
    - Lots and lots and lots of others…

MATH MODULE EXAMPLE

```
import math

radius=3.5
area=math.pi * math.pow(radius,2)
print(area)

side_a=5
side_b=7
hypotenuse=math.sqrt(side_a**2 + side_b**2)
print(hypotenuse)
```

- To Use a Module
  - **import** *module*
  - Use *dot notation:*
    - *module.function(…)*
    - *module.variable*
    - *E.g.*
      - *math.pow(2,8)*
      - *math.sqrt(25)*
      - *math.pi*

MATH MODULE EXAMPLE REVISITIED

```
from math import *

radius=3.5
area=pi * pow(radius,2)
print(area)

side_a=5
side_b=7
hypotenuse=sqrt(side_a**2 + side_b**2)
print(hypotenuse)
```

- To Use a Module
  - **from** *module* **import** *
  - Use *the function or variable name:*
    - *E.g.*
      - *pow(2,8)*
      - *sqrt(25)*
      - *pi*

APPLICATION PROGRAMMING INTERFACE (API) DOCUMENTATION

- An ***Application Programming Interface (API)*** is the collection of functions a programmer uses to interact with a code library (e.g. a module)

- ***API Documentation*** describes the functions, variables, classes and objects contained in a library.
  - math module as an example
    - Simplified: https://www.programiz.com/python-programming/modules/math
    - Definitive: https://docs.python.org/3/library/math.html