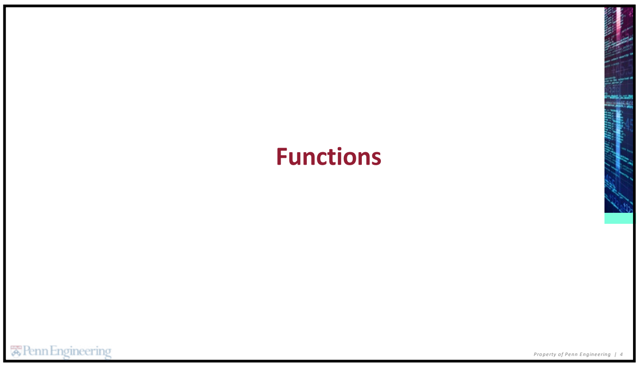
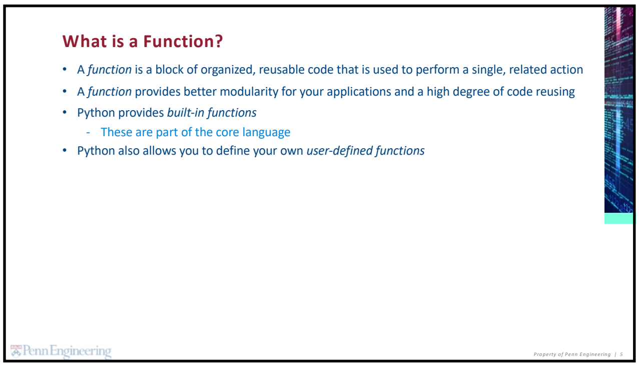


Functions

Penn Engineering

Property of Penn Engineering | 4

4



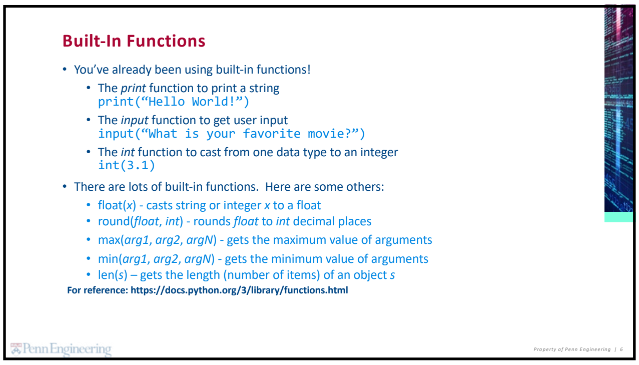
What is a Function?

- A *function* is a block of organized, reusable code that is used to perform a single, related action
- A *function* provides better modularity for your applications and a high degree of code reusing
- Python provides *built-in functions*
  - These are part of the core language
- Python also allows you to define your own *user-defined functions*

Penn Engineering

Property of Penn Engineering | 5

5



Built-In Functions

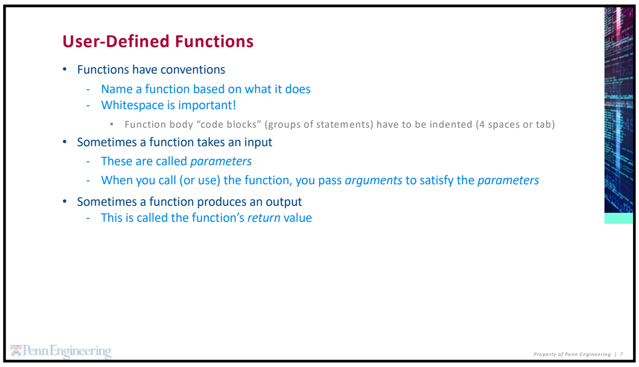
- You've already been using built-in functions!
  - The *print* function to print a string
    `print("Hello World!")`
  - The *input* function to get user input
    `input("What is your favorite movie?")`
  - The *int* function to cast from one data type to an integer
    `int(3.1)`
- There are lots of built-in functions. Here are some others:
  - float(*x*) - casts string or integer *x* to a float
  - round(*float*, *int*) - rounds *float* to *int* decimal places
  - max(*arg1*, *arg2*, *argN*) - gets the maximum value of arguments
  - min(*arg1*, *arg2*, *argN*) - gets the minimum value of arguments
  - len(*s*) – gets the length (number of items) of an object *s*

For reference: https://docs.python.org/3/library/functions.html

Penn Engineering

Property of Penn Engineering | 6

6

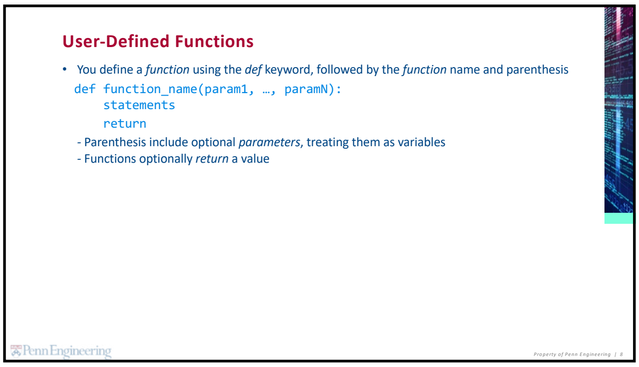


User-Defined Functions

- Functions have conventions
  - Name a function based on what it does
  - Whitespace is important!
    - Function body "code blocks" (groups of statements) have to be indented (4 spaces or tab)
- Sometimes a function takes an input
  - These are called *parameters*
  - When you call (or use) the function, you pass *arguments* to satisfy the *parameters*
- Sometimes a function produces an output
  - This is called the function's *return* value

Property of Penn Engineering | 7

7



User-Defined Functions

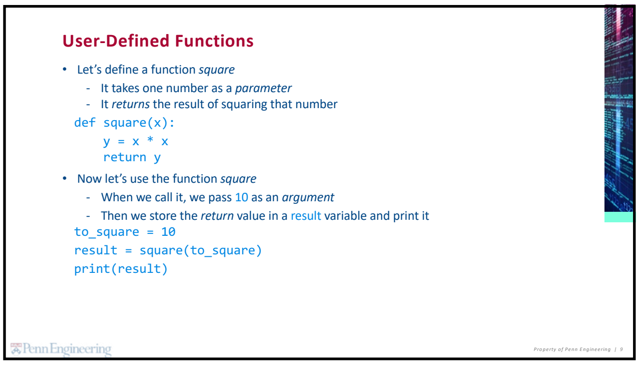
- You define a *function* using the *def* keyword, followed by the *function* name and parenthesis

```
def function_name(param1, …, paramN):
    statements
    return
```

- Parenthesis include optional *parameters*, treating them as variables
- Functions optionally *return* a value

Property of Penn Engineering | 8

8



User-Defined Functions

- Let's define a function *square*
  - It takes one number as a *parameter*
  - It *returns* the result of squaring that number

```
def square(x):
    y = x * x
    return y
```

- Now let's use the function *square*
  - When we call it, we pass 10 as an *argument*
  - Then we store the *return* value in a result variable and print it

```
to_square = 10
result = square(to_square)
print(result)
```

Property of Penn Engineering | 9

9