2.4
FUNCTIONS

unlock complex features

2.4 FUNCTIONS

- A function is a chunk of code that we can use in our programs by calling its name.
- They are handy because we can use functions over and over again.
- Python is popular because it comes with a lot of built-in functions and we already used a few.
- We will learn how to define and use our custom functions to enrich functionality in our programs.

BUILT-IN FUNCTIONS

Python comes with a lot of ___ **built-in** functions to make your life easier.

We used functions that enable us to **print** _ and accept **input** __.

We can import _____ **modules** make use of a variety of built-in functions.

POWER-UPS UNLOCKED

PRINT FUNCTIONS

You will find that there are countless ways to use print().

VARIABLE TYPE CONVERSIONS

Most useful when using the input() to prepare data for processing.

MORE STRING AND LIST FUNCTIONS

We already have seen a few, but there are more!

IMPORTING MODULES

To create games quickly we need to import various modules.

USE BUILT-IN FUNCTIONS

"output"

return

Some functions give a result which we can store in a variable.

function call

Use a function by writing the name and () which may include parameters.

```
beg = input("Enter start: ")
end = input("Enter end: ")
for count in range(beg, end):
    print(count)
```

input

parameter

Data we pass in so that the function does something with it.

Function Docs

- standard range(beg, end)
  - Returns a sequence of numbers in a list.
- standard print(s)
  - Displays a given string to the console.

WHAT YOU GONNA CALL?

Make the right function call for code snippet and sample output given.

function call

```
>>> "functions are fun".__________("fun")
```

```
2
```

Fill in the Blanks

Function Docs

- string string.count(s)

  Counts a given pattern in a string.

- string string.index(s)

  Returns index of a pattern in a string.

- string string.replace(S)

  Replaces pattern with another.

WHAT YOU GONNA CALL?

Make the right function call for code snippet and sample output given.

```
>>> name = "Jamie"
>>> print(f"Hey there {name}, ready?")
```

```
Hey there Jamie, ready?
```

format

Fill in the Blanks

Function Docs

- Standard input(prompt)

  Reads keyboard input as a string.

- standard str(object)

  Converts something into a string.

- standard print(s)

  Displays a given string to the console.

WHAT YOU GONNA CALL?

Make the right function call for code snippet and sample output given.

```
>>> guess_in = input("Enter a number: ")
>>> guess = ___int___(guess_in)
>>> if guess == random_number:
>>>     print("Wow you guessed!")
```

```
Enter a number: 6
```

Fill in the Blanks

Function Docs

- Standard float(object))

  Converts something into a float.

- standard str(object)

  Converts something into a string.

- standard int(object)

  Converts something in an integer.

WHAT YOU GONNA CALL?

Make the right function call for code snippet and sample output given.

```
>>> countdown = ["one", "two", "three"]
>>> countdown.reverse()
>>> for c in countdown
>>>     print(c + "...")
```

```
three
two
one
```

Fill in the Blanks

Function Docs

- list list.append(object)

  Add an item to the end of the list.

- list list.remove(object)

  Remove the first matching object.

- list list.reverse()

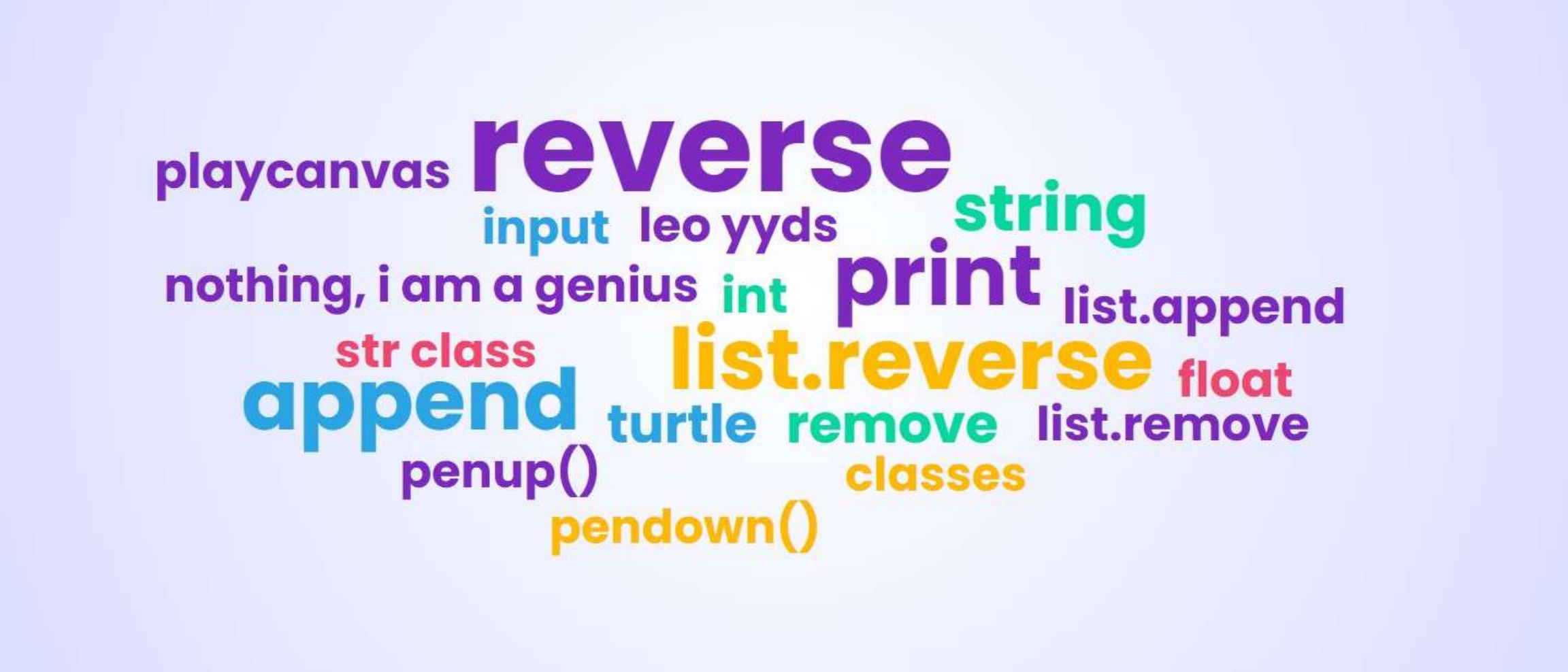
  Reverse data elements in place.

What do you think?

There are so many functions that I could not quite decide what we should use. Which ones should we practice first?

Word Cloud

Word Cloud submissions



LESSON CHALLENGE

- Time to put the theory into practice.
- You will continue to build a small component of a word search game.
- You must use all you learned so far.
- Find your tasks!

USER DEFINED FUNCTIONS

There aren't built-in functions for everything, so we need to _ define a few.

Functions should have a clear purpose and an appropriate _ name _____.

Your functions can accept _____ and return something back.

parameters

DEFINE A FUNCTION

def keyword

Your function should start with the def keyword.

unique name()

The name should summarise the purpose of the function.

```
def print_fruit_score():
    print(10)

print_fruit_score()
```

task code

The code that performs a particular task.

call function

No different to calling a built-in function.

This method does not accept parameters in ().

Did you understand?

COMPLETE THE PROGRAM

Complete the code snippet that will be part of a game. The program defines a function that displays an artistic title to make the interface more attractive for the user playing.

```
_______ print_title():
    _______ ("|* * * * * * * * * *|")
    _______ ("|* T * I * T * L * E|")
    _______("|* * * * * * * * * *|")

# call user defined function
_______()
```

Fill in the Blanks

print_title

print

def

DEFINE A FUNCTION

```
def print_fruit_score(fruit):
    if fruit == "apple":
        print(10)
    else:
        print(15)

print_fruit_score("orange")
```

argument

task code

Code in the method can be complex. Based on the type of fruit, different points are displayed.

parameters

Pass data to the function to do something with it.

We passed in the value of "orange" to the function.

Did you understand?

COMPLETE THE PROGRAM

Complete the code snippet that will be part of a game. The program defines a function that calculates and displays the total points.

```
def print_total_points(points, bonus):
    total = points + ____________
    _____________("Total: ", total)

# call user defined function
print_total_points(_____________, 500)
```

Fill in the Blanks

print

bonus

1200

DEFINE A FUNCTION

```
def get_fruit_score(fruit):
    if fruit == "apple":
        return 10
    else:
        return 15

score = get_fruit_score("orange")
```

return value

Give a value back so that it can be processed in other parts of the program.

store value

We got the value of score for "orange" and we can use it now!

Did you understand?

COMPLETE THE PROGRAM

Complete the code snippet that will be part of a game. The program defines a function that creates stars in different colours.

Fill in the Blanks

```
def get_stars(____________):
    stars = []
    for col in colours
        stars.append(col + "-star")
    ____________ stars

my_colours = ["red", "green", "blue"]
# call function and store value
____________ = get_stars(my_colours)
```

my_stars | return | colours

LESSON CHALLENGE

- Time to put the theory into practice.
- You will build something aesthetically pleasing that could look like part of a graphical game.
- You must use all you learned so far.
- Find your challenge!

Leader Board