

Exercises

Session 1 Exercises

Exercise 1.1

Now that you've run your first program, try the following:

- Change the message to anything you want
- Repeat the code on multiple lines to output several messages
- Find out what happens when you remove different parts of the code (e.g. brackets)

Don't worry if something unexpected happens. Think about what you changed and why it might have caused it to happen.

Exercise 1.2

Type these lines into your **Python console**:

```
5 - 6
8 * 9
6 / 2
5 / 0
5.0 / 2
5 % 2
2 * (10 + 3)
2 ** 4
```

What does each one do and what is its output?

Are there any outputs you didn't expect?

Exercise 1.3

In your **Python console** type each of these

```
"Cat"
"Cat" + " videos"

"Cat" * 3
"Cat" + 3
```

```
"Cat".upper()
"Cat".lower()

"the lord of the rings".title()
```

What is the output for each one and why?

One of them causes an exception. Read the exception message. What do you think it means?

Exercise 1.4

In a new Python **file** called **cat_food.py**, create a program that calculates how many cans of cat food you need to feed 10 cats

You will need:

1. A **variable** for the number of **cats**
2. A **variable** for the number of **cans** each cat eats in a day
3. A **print()** function to output the result

Extension: change the calculation to work out the amount needed for 7 days

Exercise 1.5

Rewrite **cat_food.py** to use string formatting instead of joining strings with **+**.

An example of string formatting:

```
user_name = 'sarah_1987'
age = 23

output = '{} is {} years old'.format(user_name, age)
print(output)
```

Session 2 Exercises

Exercise 2.1

Write a program that asks two questions using **input()** then prints the values that were entered. You can choose any questions that you want.

Example:

```
animal = input('Do you like dogs or cats more? ')
pet_name = input('What would name your pet? ')

print('You like {} and you would name your pet {}'.format(animal, pet_name))
```

Exercise 2.2

You have friends at your house for dinner and you've accidentally burnt the lasagne. Time to order pizza.

Write a program to calculate how many pizzas you need to feed you and your friends

```
friends = # Add input here
pizzas = friends * 0.5

print('You need {} pizzas for {} friends'.format(pizzas, friends))
```

Exercise 2.3

Create a new file called **triangle.py**. Using **turtle** draw a triangle.

A triangle has **three** sides and an angle of **120** degrees

Extension 1: Make the triangle blue

Extension 2: Draw a circle

Exercise 2.4

In this exercise you'll create a program that can draw shapes with any number of sides.

When you run the program it will ask you to input the number of sides that the shape should have. The program will then calculate the correct angle for the shape and draw it for you.

I've started the program for you:

```
import turtle

sides = int(input('Number of sides: '))

angle = 360 / sides
side_length = 60
```

```
# Add the for loop here
turtle.forward(side_length)
turtle.right(angle)

turtle.done()
```

Extension 1: Draw a spiral.

Extension 2: Write code that draws a circle using a for loop

Exercise 2.5

Create a function that draws a triangle using turtle.

Extension: Write a function that draws a circle

Exercise 2.6

Modify your triangle function so that you can set the **side length** using an argument

Extension: Use a second argument to set the **colour** of the triangle

Exercise 2.7

Complete the function to return the area of a circle

Use the comments to help you

```
def circle_area(): # add the radius argument inside the brackets
    area = 3.14 *(radius ** 2)
    # return area here

circle_1= circle_area(10)

print(circle_1)
```

Session 3 Exercises

Exercise 3.1

You have a budget of £10 and want to write a program to decide which burger restaurant to go to.

1. Input the **price** of a burger using **input()**
2. Check whether the **price** is less than or equal (**<=**) **10.00**
3. Print the result in the format below

```
Burger is within budget: True
```

Hint: remember to convert the input from a string to a decimal with **float()**

Exercise 3.2

Add code to your burger program to input whether the restaurant has a vegetarian option

The output should say whether the cost is within budget **AND** has a vegetarian option

```
Restaurant meets criteria: True
```

Extension: Add a check to see if the restaurant's rating is 3 or more

Exercise 3.3

Rewrite the output of your burger program to use if statements

If it is a good choice it should be:

```
This restaurant is a great choice!
```

If it is **not** a good choice it should be:

```
Probably not a good idea
```

Exercise 3.4

Now that you've finished your burger, you want to pay for your food. Let's write a program to calculate your meal and apply a discount if applicable.

If your total meal costs more than £20 and you have a discount, the price will be reduced by 10%. The program should print "Discount applied" or "No discount" depending on whether the discount criteria was met.

```
meal_price = float(input('How much did the meal cost? '))

discount_choice = input('Do you have a discount? y/n ')
discount_applicable = discount_choice == 'y'
```

Exercise 3.5

You're cooking a pizza and need to check that the oven is at the right temperature.

Write a program to:

- Ask the user to input the temperature
- Prints "The oven is too hot" if the temperature is over 200
- Prints "The oven is too cold" if the temperature is under 150
- Prints "The oven is at the perfect temperature" if the temperature is 180
- Prints "The temperature is close enough" for any other temperature

Extension: Ask at the start of the program whether you're cooking a cake or a pizza. The suggested temperatures for a cake should be different from the pizza.

Exercises 3.6-3.8

To practice if statements choose one of the following exercises:

- Exercise 3.6: Flip a coin
- Exercise 3.7: Rock, Scissors, Paper
- Exercise 3.8: Roulette

Exercise 3.6

This program uses random to simulate a coin flip.

To finish the program you will need to add the following:

- If the random coin flip matches the choice input by the user then they win
- Otherwise if the random coin flip does not match the choice input by the user then they lose

```
import random
```

```
def flip_coin():
```

```

random_number = random.randint(1, 2)
if random_number == 1:
    side = 'heads'
else:
    side = 'tails'
return side

choice = input('heads or tails: ')
result = flip_coin()

print('The coin landed on {}'.format(result))

```

Extension: The program should show a message to the user if they enter a choice that isn't heads or tails

Exercise 3.7

This program simulates rock, paper, scissors. The first winning condition has been added. To finish the program you'll need to add all of the other winning and losing conditions.

```

import random

def random_choice():
    choice_number = random.randint(1, 3)

    if choice_number == 1:
        choice = 'rock'
    elif choice_number == 2:
        choice = 'scissors'
    else:
        choice = 'paper'

    return choice

my_choice = input('Choose rock, scissors or paper: ')
opponent_choice = random_choice()

print('Your opponent chose {}'.format(opponent_choice))

if my_choice == 'rock' and opponent_choice == 'scissors':
    print('You win!')

```

Extension: the program should show a message to the user if they enter a choice that isn't rock, paper or scissors

Exercise 3.8

Not Quite Roulette

Ask the user to enter the following three things using **input()**:

- The amount they want to bet
- A colour (red or black)
- A number between 1 and 100

After generating a random number and colour:

- If the colour matches, the users keeps the amount that was bet
- If the number matches, the users wins double the amount that was bet
- If the colour and number matches, the users wins 100 times the amount that was bet
- When neither the colour or number matches the user wins 0
- Output the amount the user won

The following code will generate a random number and colour:

```
import random

def colour():
    random_number = random.randint(1, 2)

    if random_number == 1:
        colour = 'red'
    else:
        colour = 'black'

    return colour

random_number = random.randint(1, 100)
random_colour = colour()
```

Session 4 Exercises

Exercise 4.1

When I'm travelling in the winter I often forget to pack warm clothes. Let's write a program to help

me to remember the right clothes.

The program should check if the first item in the **clothes** list is **"shorts"**. If it is it should change the value to "warm coat".

```
clothes = [  
    "shorts",  
    "shoes",  
    "t-shirt",  
]
```

Extension: Change the other items in the list to clothing more appropriate to winter if the first item is shorts

Exercise 4.2

Make a list of game scores. Using list functions write code to output information of the scores in the following format:

```
Number of scores: 10  
Highest score: 200  
Lowest score: 3
```

Extension: Output all of the scores in descending order

Exercise 4.3

Whenever I'm shopping and I buy some bread I always forget to buy butter. Create a list and if **'bread'** is in the list, add **'butter'** to the shopping list.

Try running the program with and without bread in the list to check that your program works.

Remember the **in** operator checks if an item is in a list and the **.append()** method adds an item to a list.

Extension: only add butter to the list if it is not already in the list

Exercise 4.4

I want to work out how much money I've spent on lunch this week. I've created a list of what I spent each day.

Write a program that uses a **for** loop to calculate the total cost

```
costs = [8.30, 7.12, 5.01, 1.00, 0.99, 5.92, 3.50]
total_cost = 0
```

Extension: work out the average that I spend on lunch for the week

Exercise 4.5

Print the values of **name**, **post_code** and **street_number** from the dictionary

```
place = {
    'name': 'The Anchor',
    'post_code': 'E14 6HY',
    'street_number': '54',
    'location': {
        'longitude': 127,
        'latitude': 63,
    }
}
```

Extension: Print the values of **longitude** and **latitude** from the inner dictionary

Exercise 4.6

Using a for loop, output the values **name**, **colour** and **price** of each dictionary in the list

```
fruits = [
    {'name': 'apple', 'colour': 'red', 'price': 0.12},
    {'name': 'banana', 'colour': 'yellow', 'price': 0.2},
    {'name': 'pear', 'colour': 'green', 'price': 0.19},
]
```

Extension: Add more items to the list

Exercise 4.7

Write a program to create a random name. You should have a list of random first names and a list of last names. Choose a random item from each and display the result.

Extension: Using list of verbs and a list of nouns, create randomised sentences

Session 5 Exercises

Exercise 5.1

Create a to-do list program that writes user input to a file

The program should:

- Ask the user to input a new to-do item
- Read the contents of the existing to-do items
- Add the new to do item to the existing to-do items
- Save the updated to-do items

You will need to manually create a new file called **todo.txt** in the same folder as your program before you start

Exercise 5.2

This program is supposed to read data about trees from a file to find the shortest tree. Complete the program adding code to open **trees.csv**.

The **trees.csv** file included with your student guides. Save the csv file in the same folder as your Python program.

```
spreadsheet = # Add code to open the csv file

heights = []

for row in spreadsheet:
    tree_height = row['height']
    heights.append(tree_height)

shortest_height = min(heights)
print(shortest_height)
```

Exercise 5.3

Get the **height** and **weight** of a specific Pokemon and print the output

Extension: Print the names of all of a specific Pokemon's moves

Solutions

Session 1 Solutions

Exercise 1.2

Subtraction:

$5 - 6$

Multiplication:

$8 * 9$

Division:

$6 / 2$

Division by zero:

$5 / 0$

Float division:

$5.0 / 2$

Modulo (remainder):

$5 \% 2$

Brackets:

$2 * (10 + 3)$

Exponent (x to the power of y)

$2 ** 4$

Exercise 1.3

"Cat"

"Cat videos"

```
"CatCatCat"
```

```
Traceback (most recent call last):  
  File "<stdin>", line 1, in <module>  
TypeError: can only concatenate str (not "int") to str
```

```
"CAT"
```

```
"cat"
```

```
"The Lord Of The Rings"
```

Exercise 1.4

```
cats = 10  
cans = 2  
  
total_cans = cats * cans  
  
output = str(cats) + " cats eat " + str(total_cans) + " cans"  
print(output)
```

Extension Solution

```
cats = 10  
cans = 2  
days = 7  
  
total_cans = cats * cans * days  
  
msg = str(cats) + " cats eat " + str(total_cans) + " cans in " + str(days) + " days"  
print(msg)
```

Exercise 1.5

```
cats = 10  
cans = 2
```

```
total_cans = cats * cans
```

```
output = "{} cats eat {} cans".format(cats, total_cans)  
print(output)
```

Session 2 Solutions

Exercise 2.2

```
friends = int(input('How many friends are at your house? '))
pizzas = friends * 0.5

print('You need {} pizzas for {} friends'.format(pizzas, friends))
```

'''

Exercise 2.3

```
import turtle

side_length = 100
angle = 120

turtle.forward(side_length)
turtle.right(angle)

turtle.forward(side_length)
turtle.right(angle)

turtle.forward(side_length)
turtle.right(angle)

turtle.done()
```

Extension Solution

```
import turtle

side_length = 100
angle = 120

turtle.color('blue', 'blue')
turtle.begin_fill()

turtle.forward(side_length)
```



```
turtle.right(angle)

turtle.forward(side_length)
turtle.right(angle)

turtle.forward(side_length)
turtle.right(angle)

turtle.end_fill()

turtle.done()
```

Exercise 2.4

```
import turtle

sides = int(input('Number of sides: '))

angle = 360 / sides
side_length = 60

for side in range(sides):
    turtle.forward(side_length)
    turtle.right(angle)

turtle.done()
```

Exercise 2.5

```
import turtle

def triangle():
    side_length = 100
    angle = 120

    for side in range(3):
```

```
turtle.forward(side_length)
turtle.right(angle)

triangle()
```

Exercise 2.6

```
import turtle

def triangle(side_length):
    angle = 120

    for side in range(3):
        turtle.forward(side_length)
        turtle.right(angle)

triangle(400)
triangle(300)
triangle(200)
triangle(100)
```

Extension:

```
import turtle

def triangle(side_length, colour):
    angle = 120

    turtle.color(colour, colour)
    turtle.begin_fill()

    for side in range(3):
        turtle.forward(side_length)
        turtle.right(angle)

    turtle.end_fill()
```

```
triangle(400, 'red')
triangle(300, 'pink')
triangle(200, 'blue')
triangle(100, 'yellow')
```

Exercise 2.7

```
def circle_area(radius):
    area = 3.14 * (radius ** 2)
    return area

area = circle_area(9)

print(area)
```

'''

Session 3 Solutions

Exercise 3.1

```
price = input('How much is a burger? ')

within_budget = float(price) <= 10.00

print('Burger is within budget: {}'.format(within_budget))
```

Exercise 3.2

```
price = input('How much is a burger? ')
vegetarian = input('Is there a vegetarian option? (y/n) ')

within_budget = float(price) <= 10.00
has_vegetarian = vegetarian == 'y'

is_good_choice = within_budget and has_vegetarian

print('Restaurant meets criteria: {}'.format(is_good_choice))
```

Exercise 3.3

```
price = input('How much is a burger? ')
vegetarian = input('Is there a vegetarian option? (y/n) ')

within_budget = float(price) <= 10.00
has_vegetarian = vegetarian == 'y'

is_good_choice = within_budget and has_vegetarian

if is_good_choice:
    print('This restaurant is a great choice!')

if not is_good_choice:
    print('Probably not a good idea')
```

Exercise 3.4

```
meal_price = float(input('How much did the meal cost? '))

discount_choice = input('Do you have a discount? y/n ')
discount_applicable = discount_choice == 'y'

if discount_applicable:
    meal_price = meal_price * 0.9
    print('Discount applied')
else:
    print('No discount')

print('Total cost: {}'.format(meal_price))
```

Exercise 3.5

```
temperature = float(input('What is the temperature of the oven? '))

if temperature > 200:
    print('The oven is too hot')
elif temperature < 150:
    print('The oven is too cold')
elif temperature == 180:
    print('The oven is at the perfect temperature')
else:
    print('The temperature is close enough')
```

Exercise 3.6

```
import random

def flip_coin():
    random_number = random.randint(1, 2)
    if random_number == 1:
        side = 'heads'
    else:
        side = 'tails'
```

```

    return side

choice = input('heads or tails: ')
result = flip_coin()

print('The coin landed on {}'.format(result))

if choice == result:
    print('You win!')
else:
    print('You lose!')

```

Exercise 3.7

```

import random

def random_choice():
    choice_number = random.randint(1, 3)

    if choice_number == 1:
        choice = 'rock'
    elif choice_number == 2:
        choice = 'scissors'
    else:
        choice = 'paper'

    return choice

my_choice = input('Choose rock, scissors or paper: ')
opponent_choice = random_choice()

print('Your opponent chose {}'.format(opponent_choice))

if my_choice == 'rock' and opponent_choice == 'scissors':
    print('You win!')
elif my_choice == 'scissors' and opponent_choice == 'paper':
    print('You win!')
elif my_choice == 'paper' and opponent_choice == 'rock':
    print('You win!')
elif my_choice == opponent_choice:

```

```
    print('Draw!')
else:
    print('You lose!')
```

Exercise 3.8

```
import random

def colour():
    random_number = random.randint(1, 2)

    if random_number == 1:
        colour = 'red'
    else:
        colour = 'black'

    return colour

bet = input('Place a bet: ')
colour_choice = input('Choose a colour (red, black): ')
number_choice = input('Choose a number (1-100): ')

random_number = random.randint(1, 100)
random_colour = colour()

print('The result was {}'.format(random_number, random_colour))

if colour_choice == random_colour and number_choice != random_number:
    print('You win {}'.format(bet))
elif number_choice == random_number and colour_choice != random_colour:
    winnings = bet * 2
    print('You win {}'.format(winnings))
elif number_choice == random_number and colour_choice == random_colour:
    winnings = bet * 100
    print('You win {}'.format(winnings))
else:
    print('You lost your money')
```


Session 4 Solutions

Exercise 4.1

```
clothes = [
    "shorts",
    "shoes",
    "t-shirt",
]

if clothes[0] == 'shorts':
    clothes[0] = 'warm coat'

print(clothes)
```

Exercise 4.2

```
scores = [10, 33, 23, 19, 43, 8, 0]

print('Number of scores {}'.format(len(scores)))
print('Highest score {}'.format(max(scores)))
print('Lowest score {}'.format(min(scores)))
```

Extension

```
scores = [10, 33, 23, 19, 43, 8, 0]

print('Number of scores {}'.format(len(scores)))
print('Highest score {}'.format(max(scores)))
print('Lowest score {}'.format(min(scores)))

sorted_scores = sorted(scores)
desc_scores = list(reversed(sorted_scores))
print('All scores {}'.format(desc_scores))
```

Exercise 4.3

```
shopping_list = [
```

```
'bread',  
'cheese',  
'pop tarts',  
'carrots',  
]  
  
if 'bread' in shopping_list:  
    shopping_list.append('butter')
```

Exercise 4.4

```
costs = [8.30, 7.12, 5.01, 1.00, 0.99, 5.92, 3.50]  
total_cost = 0  
  
for cost in costs:  
    total_cost = total_cost + cost  
  
print(total_cost)
```

Exercise 4.5

```
place = {  
    'name': 'The Anchor',  
    'post_code': 'E14 6HY',  
    'street_number': '54',  
    'location': {  
        'longitude': 127,  
        'latitude': 63,  
    }  
}  
  
print(place['name'])  
print(place['post_code'])  
print(place['street_number'])
```

Exercise 4.6

```
fruits = [
```

```
{'name': 'apple', 'colour': 'red', 'price': 0.12},  
{'name': 'banana', 'colour': 'yellow', 'price': 0.2},  
{'name': 'pear', 'colour': 'green', 'price': 0.19},  
]
```

```
for fruit in fruits:  
    print(fruit['name'])  
    print(fruit['colour'])  
    print(fruit['price'])
```

Exercise 4.7

```
import random  
  
first_names = ['Dierdre', 'Patricia', 'Edelbert']  
last_names = ['Johnson', 'Davis', 'Oak']  
  
first_name = random.choice(first_names)  
last_name = random.choice(last_names)  
  
print('{} {}'.format(first_name, last_name))
```

Session 5 Solutions

Exercise 5.1

```
new_item = input('Enter a to-do item: ')

with open('todo.txt', 'r') as todo_file:
    todo = todo_file.read()

todo = todo + new_item + '\n'

with open('todo.txt', 'w+') as todo_file:
    todo_file.write(todo)
```

Exercise 5.2

```
import csv

with open('trees.csv', 'r') as csv_file:
    spreadsheet = csv.DictReader(csv_file)

    heights = [ ]

    for row in spreadsheet:
        tree_height = row['height']
        heights.append(tree_height)

shortest_height = min(heights)
print(shortest_height)
```

Exercise 5.3

```
import requests

pokemon_number = input("What is the Pokemon's ID? ")

url = 'https://pokeapi.co/api/v2/pokemon/{}/'.format(pokemon_number)

response = requests.get(url)
```

```
pokemon = response.json()
```

```
print(pokemon['name'])
```

```
print(pokemon['height'])
```

```
print(pokemon['weight'])
```