

Starter: Rewrite this code to use a for loop and the `range()` function:

```
print('~' * 0)
print('~' * 1)
print('~' * 2)
print('~' * 3)
print('~' * 4)
print('~' * 5)
print('~' * 6)
print('~' * 7)
print('~' * 8)
```

Starter solution

```
for number in range(9):  
    print('~' * number)
```

```
~  
~~  
~~~  
~~~~  
~~~~~  
~~~~~  
~~~~~  
~~~~~  
~~~~~
```



Python Session 3

Topics in this session:

1. Comparison Operators
2. Logical Operators
3. If Statements

Comparisons and Logical Operators

Boolean: A data-type that is either True or False

Comparison operator: compare values to determine whether something is True or False

This code checks if the user has input 'Monday' using the == operator

```
today = input('What day is it? ')
is_monday = today == 'Monday'
print('Today is Monday: {}'.format(is_monday))
```

```
What day is it? Monday
Today is Monday: True
```

Summary of comparison operators in Python

Name	Python
Equal to	==
Not equal	!=
Greater than	>
Less than	<
Greater than or equal	>=
Less than or equal	<=

`float()` can convert strings to floats

This code checks if the current temperature is freezing:

```
temperature = input('What is the temperature? ')
is_freezing = float(temperature) <= 0.0
print('The temperature is freezing: {}'.format(is_freezing))
```

What is the temperature? 18

The temperature is freezing: False

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 ()`

Solution

```
price = input('How much is a burger? ')
within_budget = float(price) <= 10.00
print('Burger is within budget: {}'.format(within_budget))
```

```
How much is a burger? 8.99
Burger is within budget: True
```

There are logical operators to combine multiple checks

Python	What it does
and	both expressions are True
or	at least one expression is True
not	reverse the expression (True becomes False and vice-versa)

This program will work out if you should visit Mars based on whether you want to visit and if you can afford it:

```
mars_choice = input('Would you like to visit Mars? y/n ')
is_willing = mars_choice == 'y'

affordable = input('Can you afford to visit Mars? y/n ')
can_afford = affordable == 'y'

should_visit_mars = is_willing and can_afford

print('You should visit Mars: {}'.format(should_visit_mars))
```

```
Would you like to visit Mars? y/n y
Can you afford to visit Mars? y/n n
You should visit Mars: False
```

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
```

Solution:

```
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))
```

```
How much is a burger? 9.99
Is there a vegetarian option? (y/n) y
Restaurant meets criteria: True
```

If Statements

if statement: used to run a block of code depending on whether a condition is True or False

```
password = input('password: ')\n\nif password == 'jumanji':\n    print('Success!')
```

```
password: jumanji\nSuccess!
```

An `if` statement has the following:

1. The `if` keyword
2. A condition (comparison)
3. A colon
4. Body (indented four spaces)

condition

body

```
if house_number = 12:  
    print('this is the house')
```

This program checks whether you are an admin and you have entered the right password:

```
name = input("What is your name? ")
is_admin = name == 'admin'

password = input("What is your password? ")
is_password_correct = password == 'dinosaurs'

if is_admin and is_password_correct:
    print('Welcome')

if not is_admin or not is_password_correct:
    print('Go away')
```

```
What is your name? admin
What is your password? tigers
Go away
```

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

Solution

```
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')
```

```
How much is a burger? 9.99
Is there a vegetarian option? (y/n) y
This restaurant is a great choice!
```

Else Statements

else statement: Used with an `if` statement and will run when the `if` condition is `False`

```
password = input('password: ')\n\nif password == 'jumanji':\n    print('Success!')\nelse:\n    print('Failure!')
```

```
password: cluedo\nFailure!
```


Here's the admin program rewritten to use `else` :

```
name = input("What is your name? ")
is_admin = name == 'admin'

password = input("What is your password? ")
is_password_correct = password == 'dinosaurs'

if is_admin and is_password_correct:
    print('Welcome')
else:
    print('Go away')
```

```
What is your name? admin
What is your password? tigers
Go away
```

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'
```

Solution

```
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))
```

```
How much did the meal cost? 30
Do you have a discount? y/n y
Discount applied
Total cost: 27.0
```

Elif Statements

elif statement: used after `if` statements to check whether another condition is `True` or `False`

```
dog_size = int(input('How big is the dog? '))

if dog_size > 75:
    print('That is a big dog')

elif dog_size < 25:
    print('That is a small dog')

else:
    print('That is an average dog')
```

How big is the dog? 10
That is a small dog

You can use multiple `elif` statements together

```
dog_size = int(input('How big is the dog? '))  
  
if dog_size > 75:  
    print('That is a big dog')  
  
elif dog_size < 10:  
    print('That dog could fit in my pocket')  
  
elif dog_size < 25:  
    print('That is a small dog')  
  
else:  
    print('That is an average dog')
```

How big is the dog? 6
That dog could fit in my pocket

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

Solution

```
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')
```

What is the temperature of the oven? 170
The temperature is close enough

Random

Python has a built-in library for random data

```
import random  
  
random_integer = random.randint(1, 100)  
  
print(random_integer)
```

50

The `randint()` function generates a random number between two values

This program uses `randint()` to simulate dice with any number of sides

```
import random

sides = int(input('How many sides does the die have? '))
random_integer = random.randint(1, sides)

print('You rolled a {}'.format(random_integer))
```

```
How many sides does the die have? 6
You rolled a 2
```

To practice if statements choose one of the following exercises in your student guide:

- 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))
```

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!')
```

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()
```


Recap

This session:

1. Comparison operators
2. Logical Operators
3. If Statements

Question 1: Equals to (==) is a comparison operator. Name two more comparison operators

Question 2: What is the output of this code?

```
print(True and True)
print(True and False)
print(True or True)
print(True or False)
```

Question 3: I expect this code to output "This is too many apples", but instead it outputs "That is a sensible number of apples". Why does this happen?

```
apples = 100

if apples >= 10:
    print('That is a sensible number of apples')
elif apples > 50:
    print('This is too many apples')
elif apples < 10:
    print('That is not enough apples')
```

Homework: Session 3 homework questions in your student guide