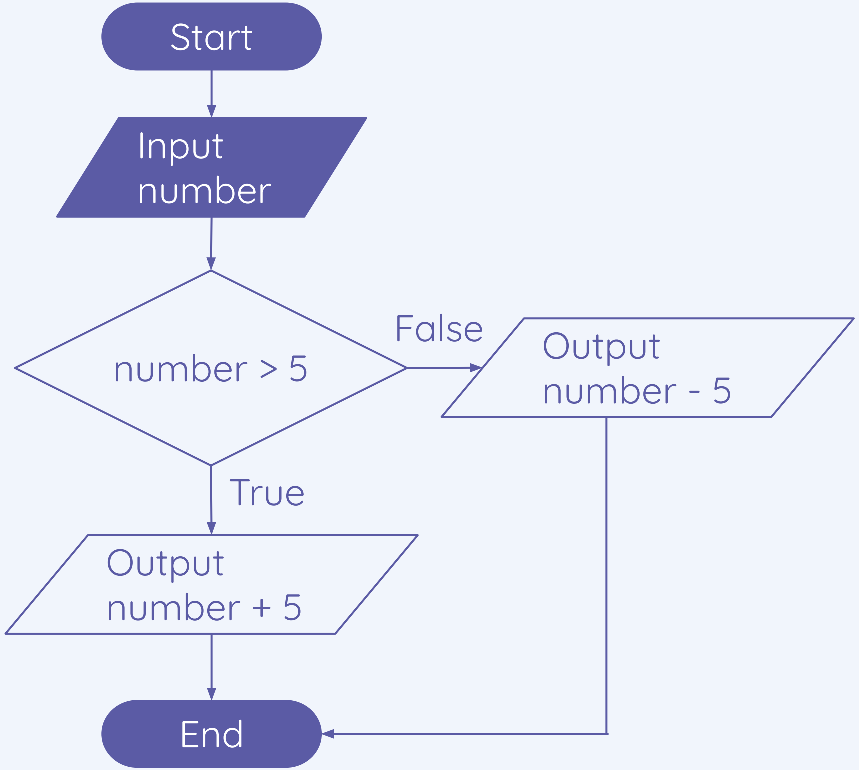
Lesson 6: Selection

Make a prediction

Question

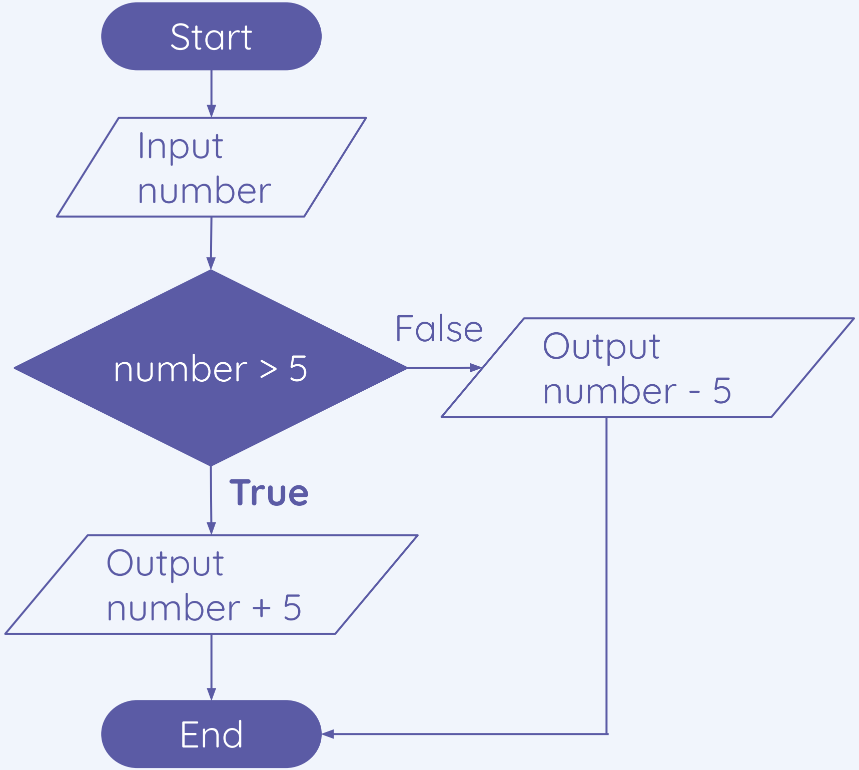
If you **input** the number 10, what will be the **output**?



Make a prediction

10 is more than 5

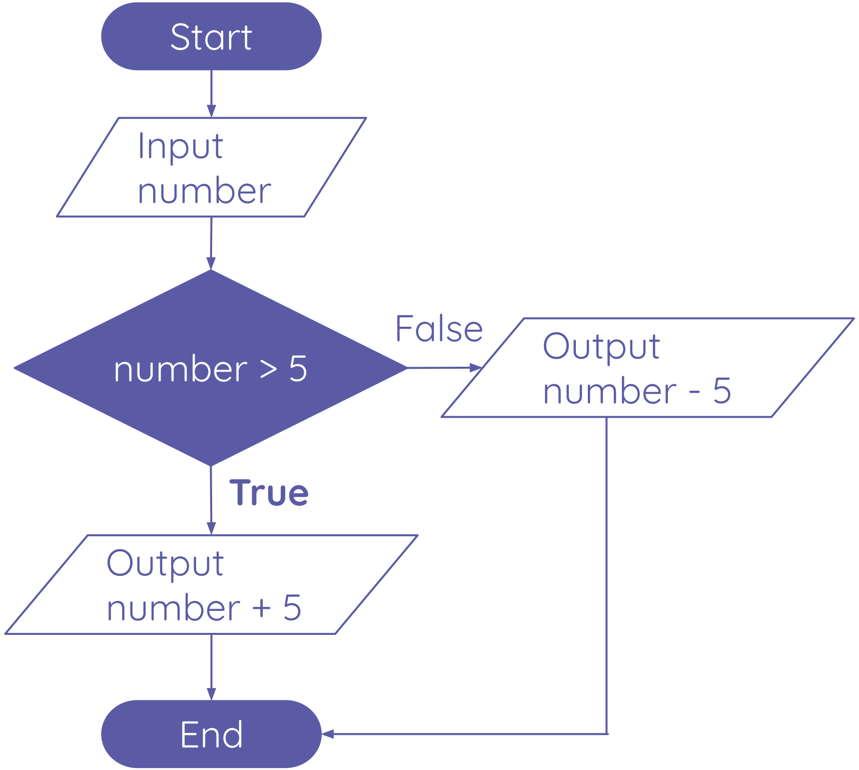
So this will be True



Make a prediction

10 is more than 5

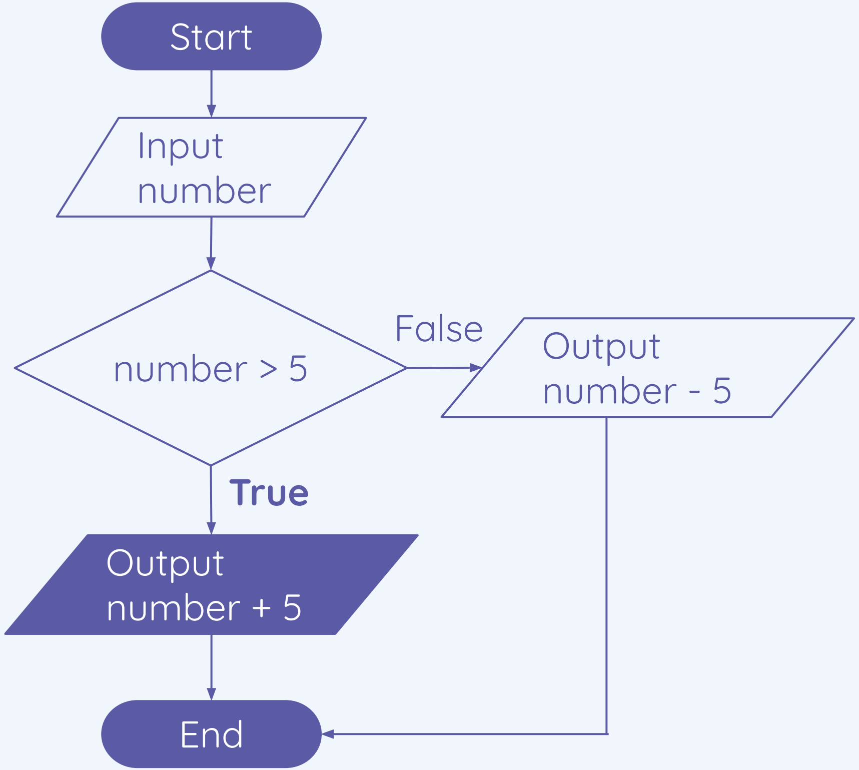
So this will be True



Make a prediction

10 + 5 is 15

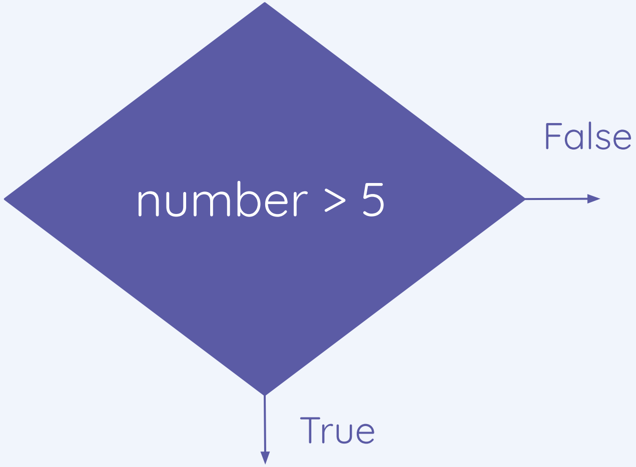
So the output will be **15**



Make a prediction

The decision symbol is used in a flowchart to **control the flow of execution**.

Inside the diamond is a **condition**. If the **condition** is **True**, then during program execution it will follow one path. If it is **False**, it will follow the other path.



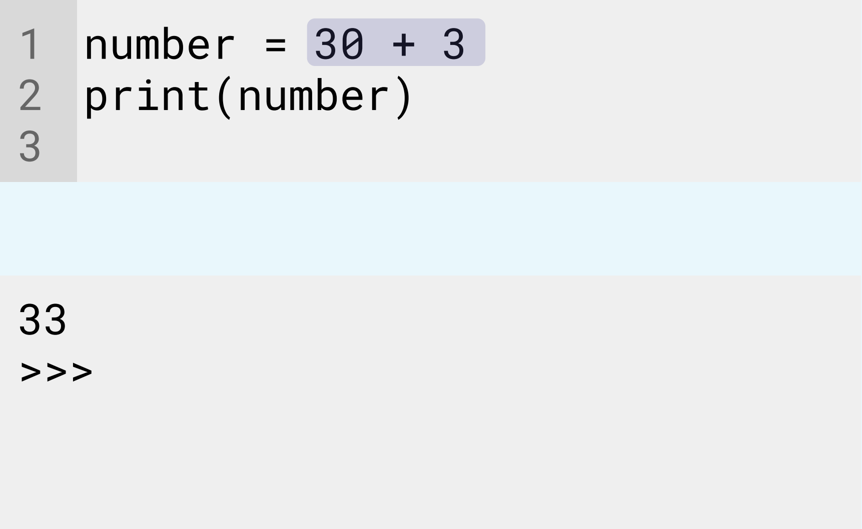
Lesson 8: Selection

In this lesson, you will:

- Identify flowchart symbols and describe how to use them (decision)
- Define a condition as an expression that can be evaluated to either True or False
- Identify that selection uses conditions to control the flow of execution
- Walk through code that includes selection (`if`, `elif`, `else`)

What is a condition?

You already know that **arithmetic expressions** evaluate to a number.



```
number = 30 + 3
print(number)

```

```
33
>>>
```

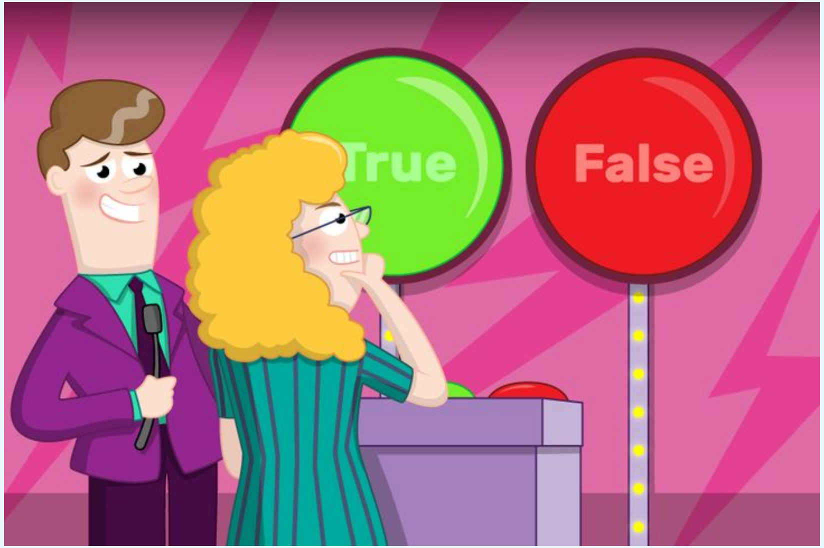
What is a condition?

Logical expressions evaluate to **True** or **False**.

```
score = 20
you_won = score > 30
print(you_won)
```

```
False
>>>
```

Will the following logical expressions evaluate to True or False?



3 > 4

Comparison operators

==	equal to
<	less than
>	more than

False

$$(3+10) < (3*10)$$

Comparison operators

==	equal to
<	less than
>	more than

True

$$(5-2+10/2) == 8$$

Comparison operators

==	equal to
<	less than
>	more than

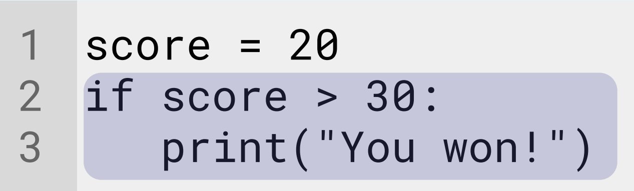
True

Selection

```
score = 20
if score > 30:
    print("You won!")
```

You can use a **logical expression** in a **condition** to control the flow of execution in your programs.

Selection



```python
score = 20
if score > 30:
    print("You won!")
```

You can use a **logical expression** in a **condition** to control the flow of execution in your programs.

Here a condition has been used in a **selection statement**.

This is also known as an **if statement**.

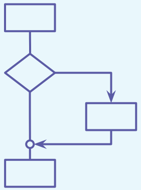
Selection

```python
score = 20
if score > 30:
    print("You won!")
```

Selection statements control the flow of execution because a **block** of code will only run if the condition is **True**.

Selection: walkthrough

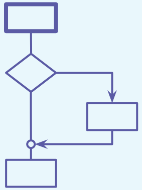
```
score = 20
if score > 30:
    print("You won!")
print("The end")
```



Here is a walkthrough of a selection statement.

Selection: walkthrough

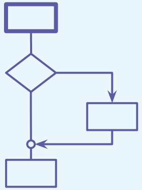
```
score = 20
if score > 30:
    print("You won!")
print("The end")
```



score is assigned 20

Selection: walkthrough

```python
score = 20
if score > 30:
    print("You won!")
print("The end")
```



State

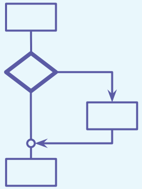
score 20

score is assigned 20

Selection: walkthrough

```
score = 20
if score > 30:
    print("You won!")
print("The end")
```

False



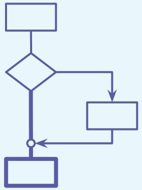
State

score	20

`score` is not greater than `30`, so the condition is **False**

Selection: walkthrough

```
1  score = 20
2  if score > 30:
3      print("You won!")
4  print("The end")
```



State

score `20`

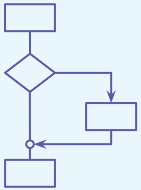
Output

```
The end
```

The flow of control moves to line 4 and outputs **The end**

Selection: walkthrough

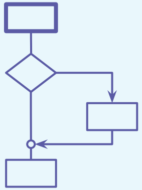
```
score = 35
if score > 30:
    print("You won!")
print("The end")
```



This time, `score` is assigned 35

Selection: walkthrough

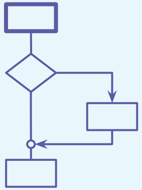
```
score = 35
if score > 30:
    print("You won!")
print("The end")
```



This time, `score` is assigned `35`

Selection: walkthrough

```
score = 35
if score > 30:
    print("You won!")
print("The end")
```



State

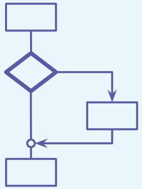
score 35

This time, **`score`** is assigned **35**

Selection: walkthrough

```
score = 35
if score > 30:
    print("You won!")
print("The end")
```

True



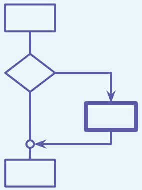
State

score 35

score is over 30 so this condition is now **True**

Selection: walkthrough

```
1 score = 20
2 if score > 30:
3     print("You won!")
4 print("The end")
```



State

score 35

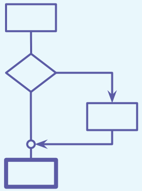
Output

```
You won!
```

The flow of execution moves to line 3 and outputs `You won!`

Selection: walkthrough

```
1  score = 20
2  if score > 30:
3      print("You won!")
4  print("The end")
```



State

```
score    35
```

Output

```
You won!
The end
```

The flow of execution then carries on to line 4 and outputs **The end**

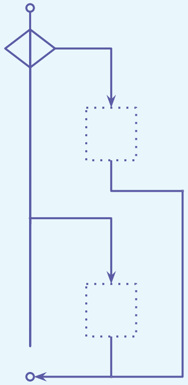
Another branch

You can provide another **branch** to your **selection statement** by using `if-else`.

When you use `if-else` you are saying:

“If this condition is True then do this. ***Else****, do this.”*

```
if this_is_true:
    do this
else:
    do this
```



An example of using else

```
print("What’s your name?")
user = input()

if user == "Elizabeth":
  print("Good morning Your Majesty")
else:
  print("Hello", user)
```

The condition will check if the value of `user` is equal to the string `"Elizabeth"`.

The expression `user == "Elizabeth"` will evaluate to either `True` or `False`.

This is the `if`-block, i.e. the code that will be executed if the condition is `True`.

This is the `else`-block, i.e. the code that will be executed if the condition is `False`.

Only **one** of these blocks will be executed, depending on the value of the condition.

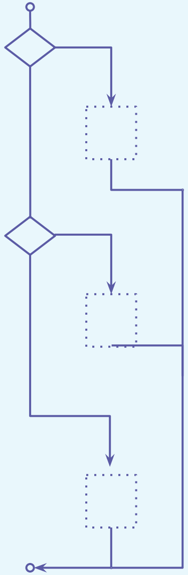
More branches!

You can provide further **branches** by using `elif`.

When you use `if-elif-else` you are saying:

“If this condition is True then do this. Else if this condition is True then do this. Else, do this.”

```
if this_is_true:
   do this
elif this_is_true:
   do this
else:
   do this
```



An example of using `elif`

```
print("What’s your name?")
user = input()
if user == "Elizabeth":
    print("Good morning Your Majesty")
elif user == "Alan Sugar":
    print("Good morning Sir")
else:
  print("Hello", user)
```

If the expression `user == "Elizabeth"` is evaluated to `False`.

The `elif` condition will check if the value of `user` is equal to the string `"Alan Sugar"`.

If this is `True,` the `elif`-block will be executed.

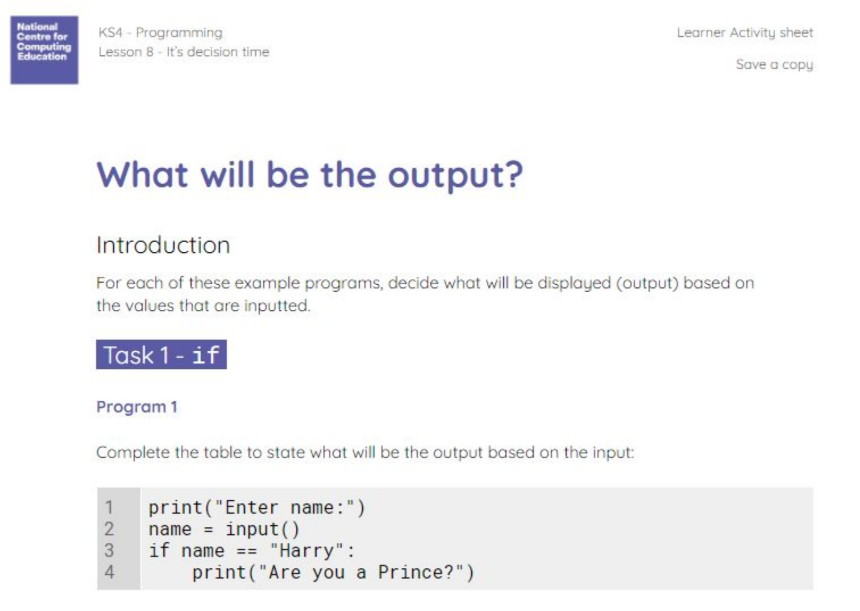
The `else`-block will be executed if both conditions are `False`.

Only **one** of these blocks will be executed, depending on the value of the conditions.

Activity 1

What will be the output?

Use the **worksheet** to decide what will be the output based on the input.



KS4 - Programming
Lesson 8 - It's decision time

Learner Activity sheet

Save a copy

What will be the output?

Introduction

For each of these example programs, decide what will be displayed (output) based on the values that are inputted.

Task 1 - `if`

Program 1

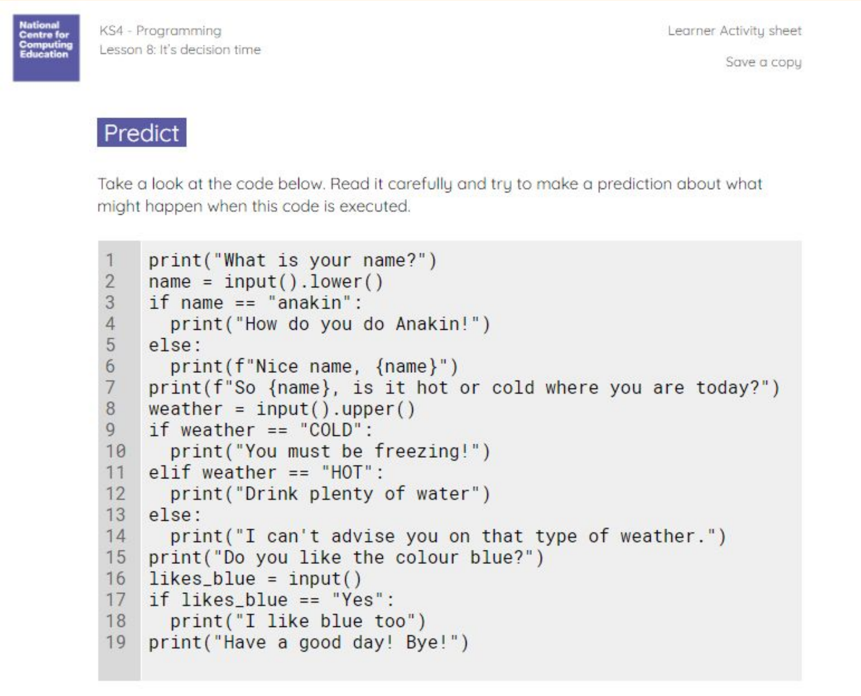
Complete the table to state what will be the output based on the input:

```
print("Enter name:")
name = input()
if name == "Harry":
    print("Are you a Prince?")
```

Activity 2

Chatterbot

Use the worksheet to **investigate** and **modify** a chatterbot.



National Centre for Computing Education

KS4 - Programming
Lesson 8: It's decision time

Learner Activity sheet

Save a copy

Predict

Take a look at the code below. Read it carefully and try to make a prediction about what might happen when this code is executed.

```
print("What is your name?")
name = input().lower()
if name == "anakin":
  print("How do you do Anakin!")
else:
  print(f"Nice name, {name}")
print(f"So {name}, is it hot or cold where you are today?")
weather = input().upper()
if weather == "COLD":
  print("You must be freezing!")
elif weather == "HOT":
  print("Drink plenty of water")
else:
  print("I can't advise you on that type of weather.")
print("Do you like the colour blue?")
likes_blue = input()
if likes_blue == "Yes":
  print("I like blue too")
print("Have a good day! Bye!")
```

Multiple choice questions

```
print("Name the best programming language?")
language = input()
if language == "Python":
    print("Hello Pythonista")
else:
    print("Wrong, Python is the best!")
```

Questions

When this program is executed, what will be displayed on the screen, if the user enters `Python` at the prompt?

1. `Wrong, Python is the best!`
2. [✓ `Hello Pythonista`]
3. `Hello, Pythonista`
`Wrong, Python is the best!`
4. There is an error in the program

Multiple choice questions

```
print("Enter a number")
number = int(input())
if number > 0:
    print(f"{number}, is positive")
else:
    print(f"{number}, is negative")
```

Questions

When this program is executed, what will be displayed on the screen if the user enters `0` at the prompt?

1 `0 is positive`
`0 is negative`

2 `0 is negative`

3 `0 is positive`

4 There is an error in the program

Multiple choice questions

```
number = 13
if number < 10:
    print("small")
elif number < 100:
    print("medium")
elif number < 1000:
    print("large")
```

Questions

When this program is executed, what will be displayed on the screen?

1 `small`

[✓ 2] `medium`

3 `medium`
`large`

4 `large`

Next lesson

In this lesson, you…

Discovered how to use conditions to control the flow of execution in your programs

Next lesson, you will…

Make your own programs that use selection