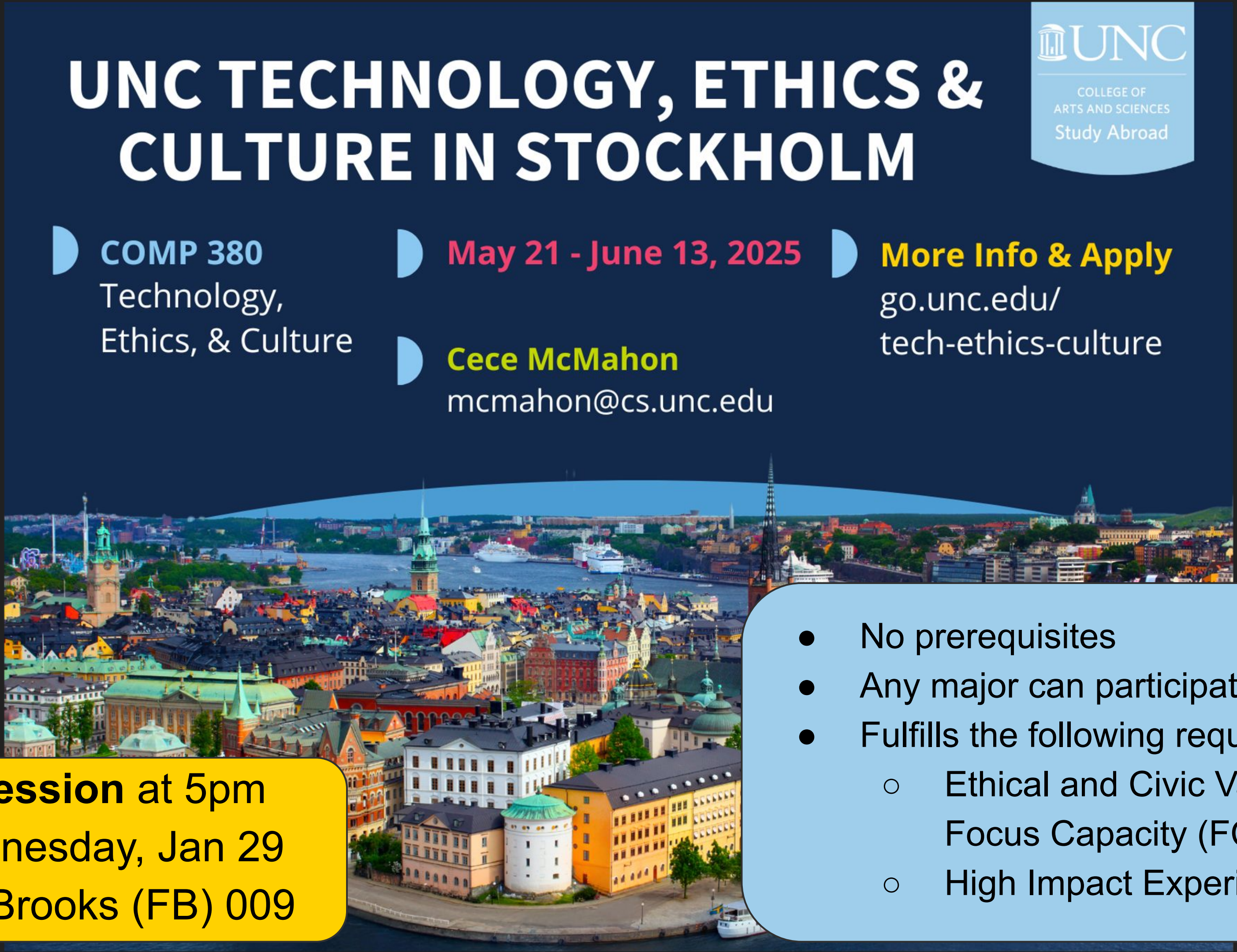


UNC TECHNOLOGY, ETHICS & CULTURE IN STOCKHOLM

UNC
COLLEGE OF
ARTS AND SCIENCES
Study Abroad

COMP 380
Technology,
Ethics, & Culture

May 21 - June 13, 2025

Cece McMahon
mcmahon@cs.unc.edu

More Info & Apply
go.unc.edu/
tech-ethics-culture

Info Session at 5pm
on Wednesday, Jan 29
in Fred Brooks (FB) 009

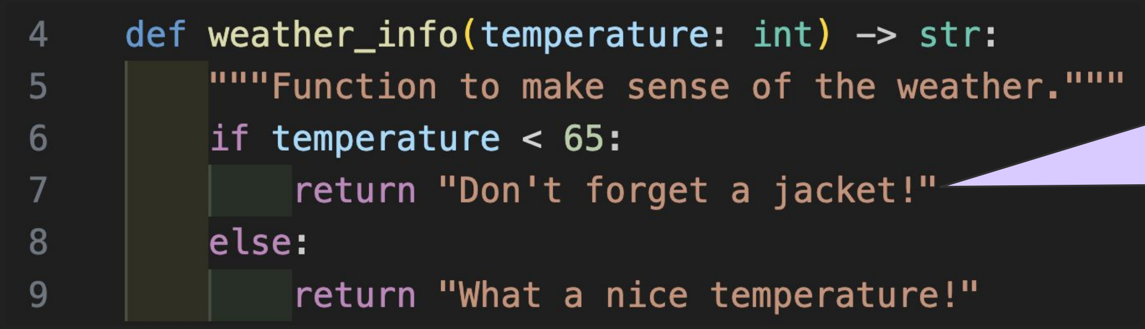
- No prerequisites
- Any major can participate!
- Fulfills the following requirements:
  - Ethical and Civic Values Focus Capacity (FC-Values)
  - High Impact Experience

COMP110

CL07 - Conditional Control Flow

If-then-else / *Conditional* Statements

Code that behaves conditionally based on input values



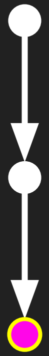
```python
def weather_info(temperature: int) -> str:
    """Function to make sense of the weather."""
    if temperature < 65:
        return "Don't forget a jacket!"
    else:
        return "What a nice temperature!"
```

Remember: as soon as we execute a return statement, we exit the function body and send the return value (RV) to the return address (RA)

Predict what the following function calls will evaluate to:

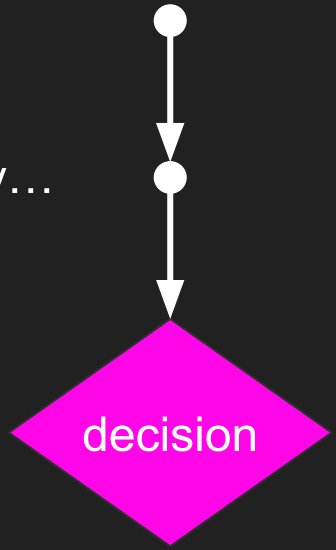
1. `weather_info(temperature=10)`
2. `weather_info(temperature=65)`
3. `weather_info(temperature=93)`

Control flow is *linear*



Going about your day…

Control flow is *linear*



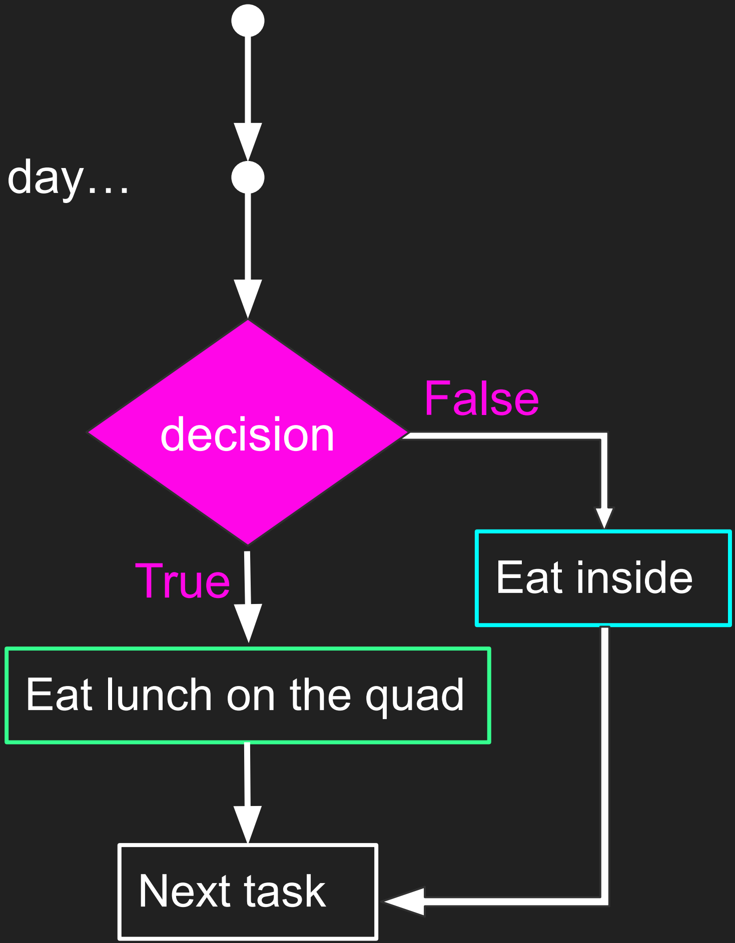
Going about your day…

Is the weather nice?

Control flow is *linear*

Going about your day…

Is the weather nice?



decision

False

True

Eat inside

Eat lunch on the quad

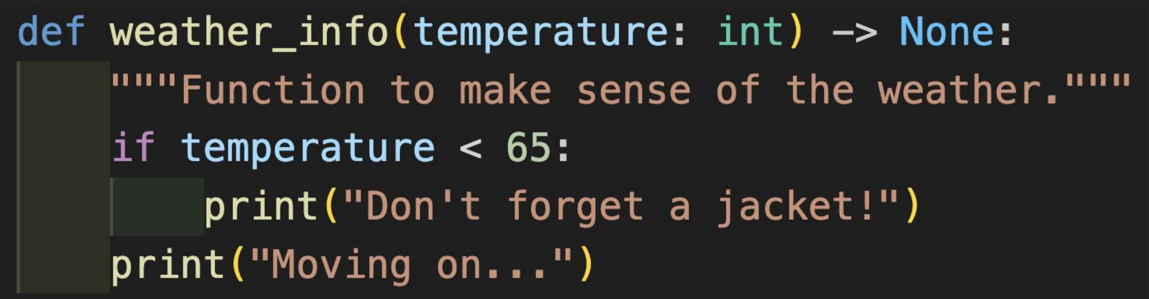
Next task

If-then / *Conditional* Statements

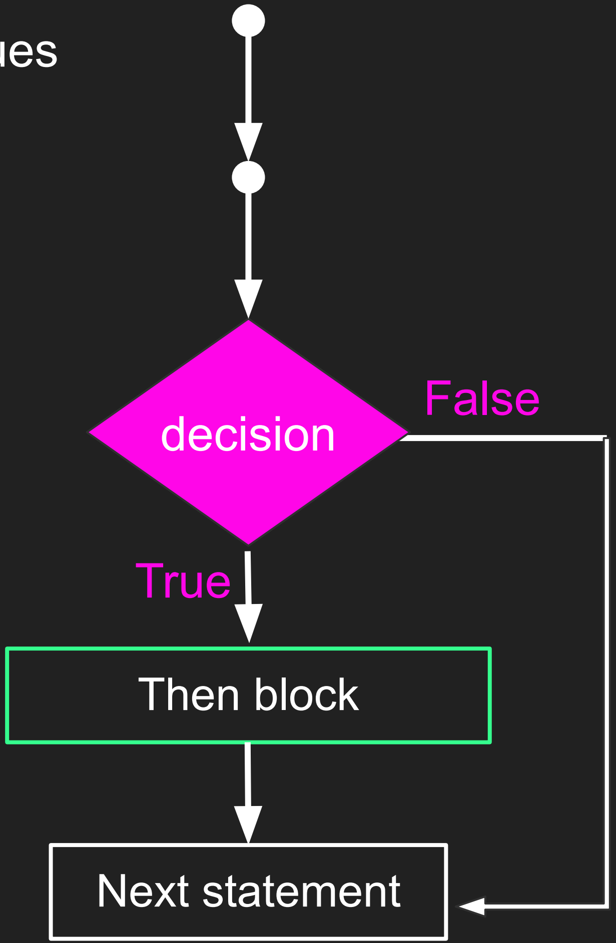
Code that behaves conditionally based on input values

bool

```
if <condition>:

    <then, execute these statements>

<rest of program>
```



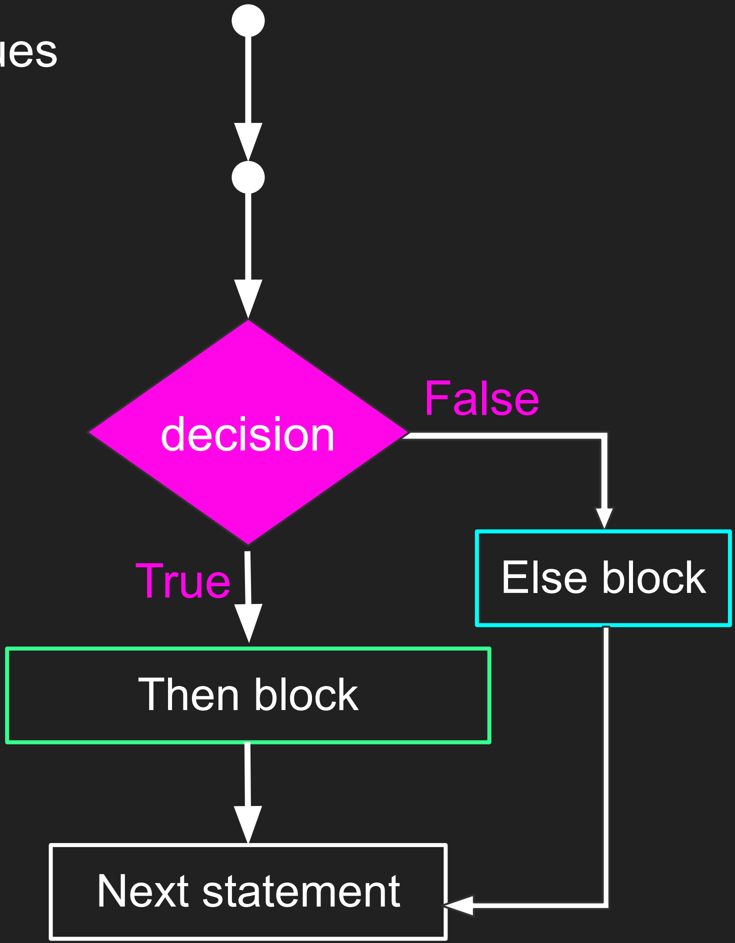
```
def weather_info(temperature: int) -> None:
    """Function to make sense of the weather."""
    if temperature < 65:
        print("Don't forget a jacket!")
    print("Moving on...")
```



If-then-else / *Conditional* Statements

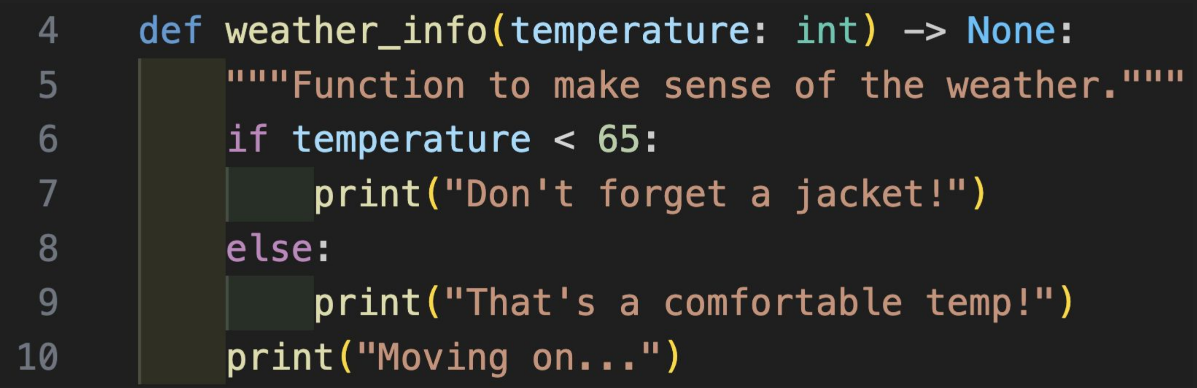
Code that behaves conditionally based on input values

```
if <condition>:
    <then, execute these statements>
else:
    <execute these other statements>
<rest of program>
```



If-then-else / *Conditional* Statements

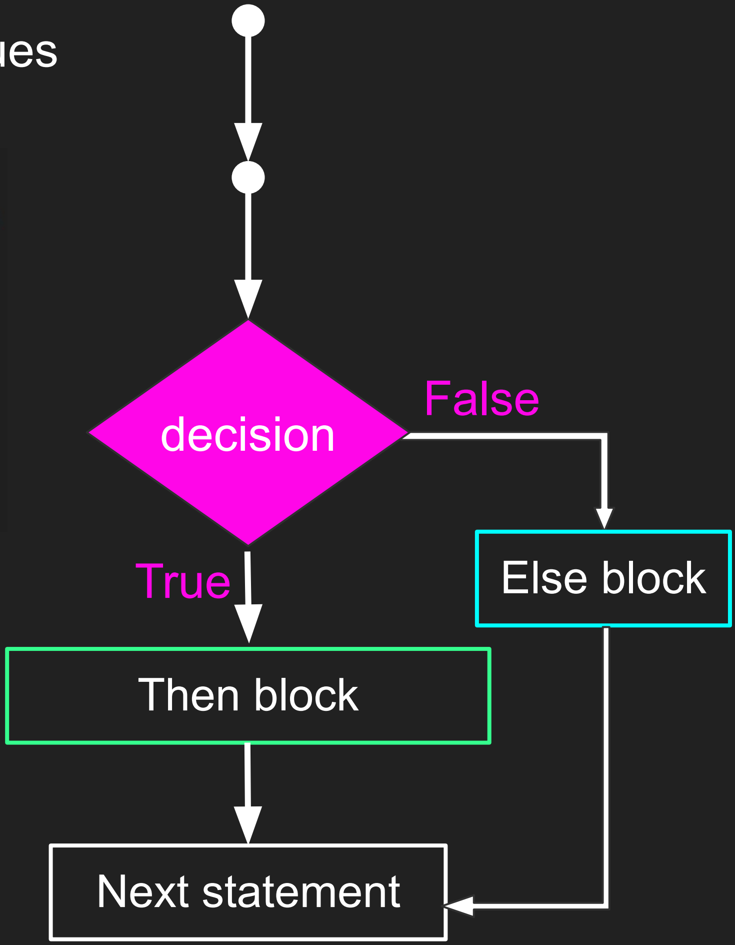
Code that behaves conditionally based on input values



```
4   def weather_info(temperature: int) -> None:
5       """Function to make sense of the weather."""
6       if temperature < 65:
7           print("Don't forget a jacket!")
8       else:
9           print("That's a comfortable temp!")
10      print("Moving on...")
```

Determine which lines would execute as a result of this function call: **`weather_info(temperature=63)`**

And this one:
`weather_info(temperature=75)`

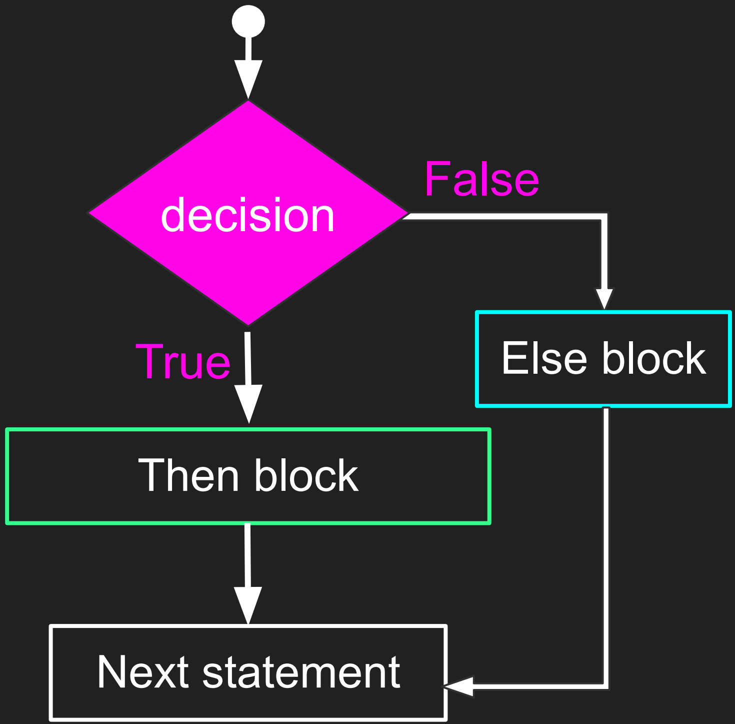


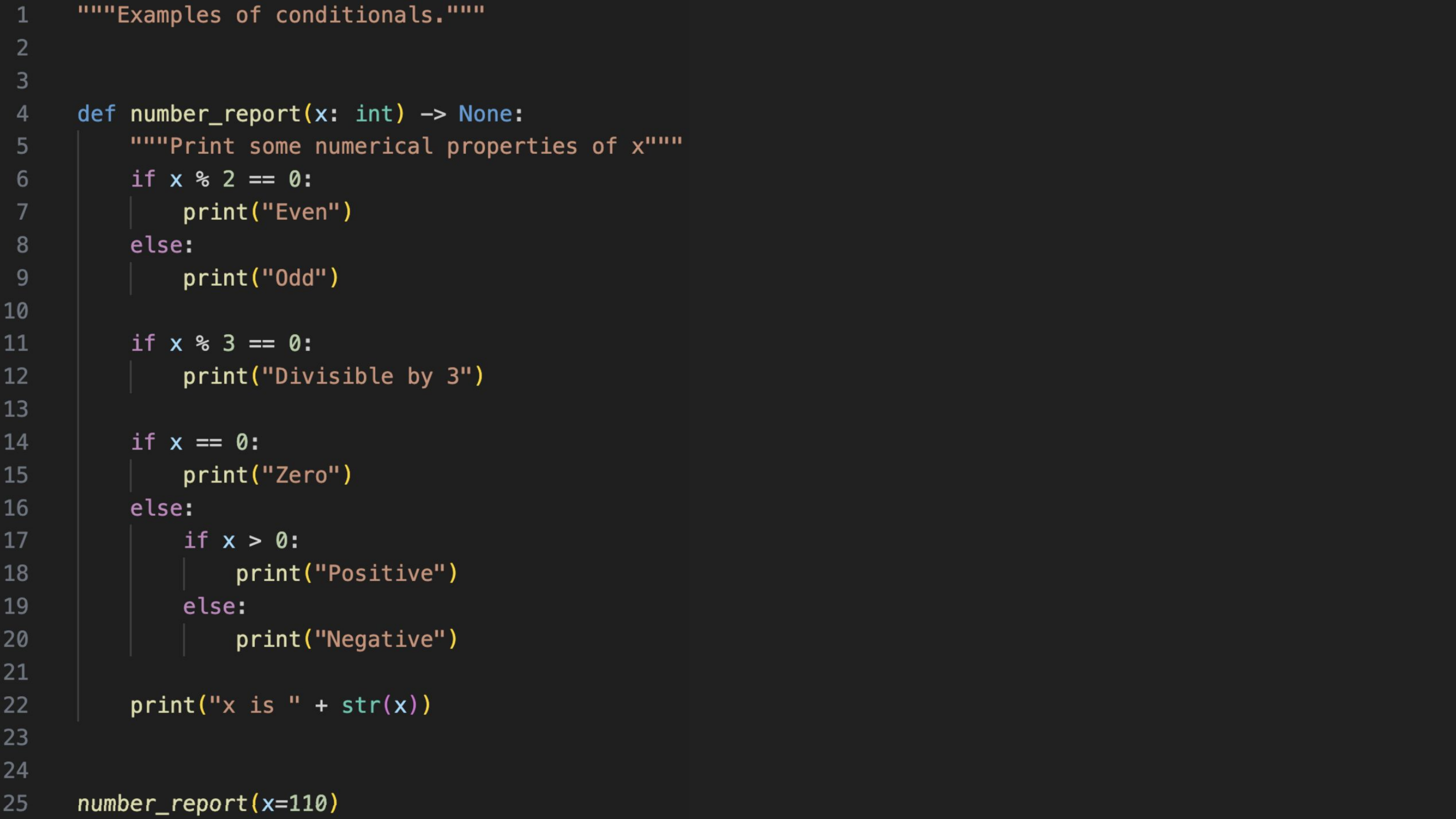
General syntax and semantics

```
if <condition>:
        <then block>
else:
        <else block>
<rest of program>
```

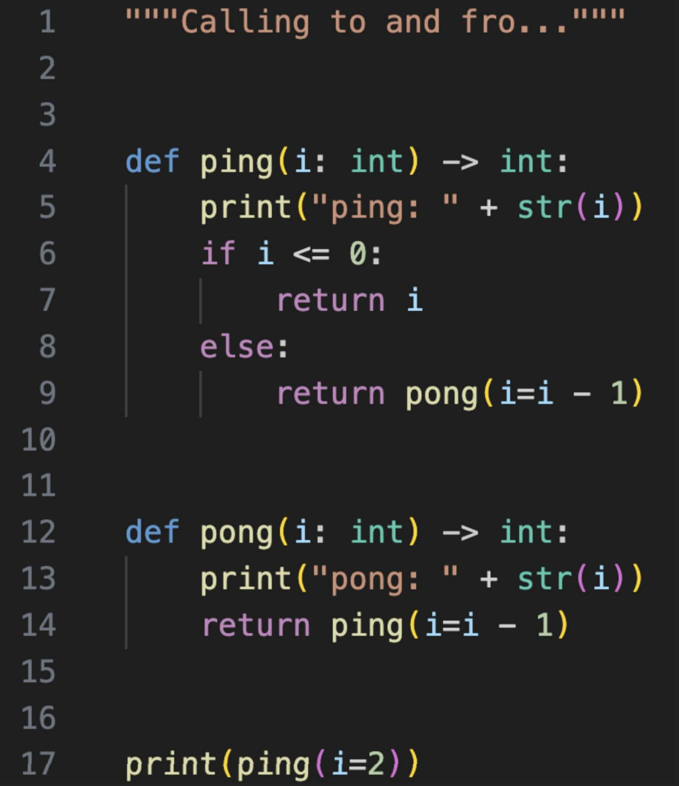
Semantics:

1. When evaluation reaches an **if statement**, the **boolean test expression** is evaluated.
2. If the expression evaluates to `True`, control continues into the **then statement block**. If the then statement block completes without a return, control continues by moving on to the next statement after the if statement.
3. Otherwise, if the test expression evaluates to `False`, control *jumps over the then block* and continues to the next line, whether it is an **else statement block** (if there is one) or the next statement in the program.

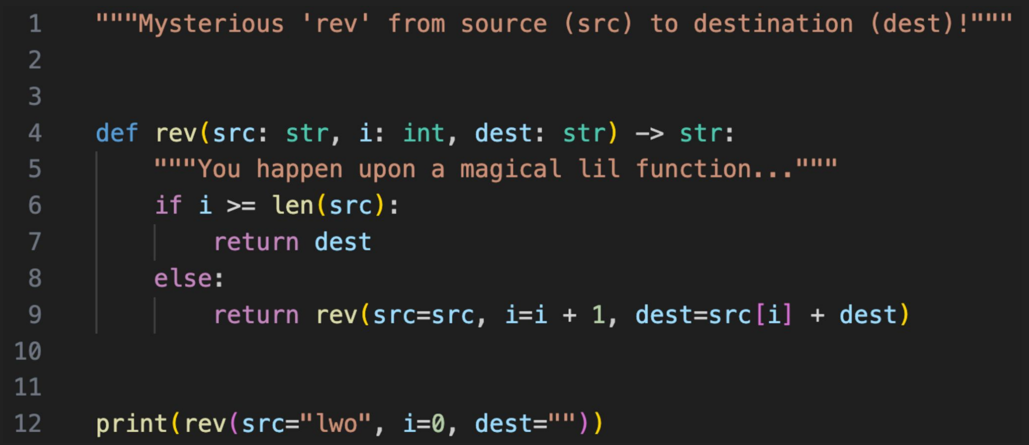




```python
"""Examples of conditionals."""


def number_report(x: int) -> None:
    """Print some numerical properties of x"""
    if x % 2 == 0:
        print("Even")
    else:
        print("Odd")

    if x % 3 == 0:
        print("Divisible by 3")

    if x == 0:
        print("Zero")
    else:
        if x > 0:
            print("Positive")
        else:
            print("Negative")

    print("x is " + str(x))


number_report(x=110)
```



```python
"""Calling to and fro..."""


def ping(i: int) -> int:
    print("ping: " + str(i))
    if i <= 0:
        return i
    else:
        return pong(i=i - 1)


def pong(i: int) -> int:
    print("pong: " + str(i))
    return ping(i=i - 1)


print(ping(i=2))
```



```python
"""Mysterious 'rev' from source (src) to destination (dest)!"""


def rev(src: str, i: int, dest: str) -> str:
    """You happen upon a magical lil function..."""
    if i >= len(src):
        return dest
    else:
        return rev(src=src, i=i + 1, dest=src[i] + dest)


print(rev(src="lwo", i=0, dest=""))
```

Practice

Write a function called check_first_letter that takes a input two strs: word and letter

It should return “match!” if the first character of word is letter

Otherwise, it should return “no match!”

Examples:

- check_first_letter(word=”happy”, letter=”h”) would return “match!”
- check_first_letter(word=”happy”, letter=”s”) would return “no match!”