

# Control: loops

## Recap

- `if / elif / else`
- comparison operators ( `>` , `>=` , ... )
- logical operators ( `and` , `or` , `not` )
- boolean values & expressions ( `2 > 1` , `True` , `False` )
- control flow usign `return`

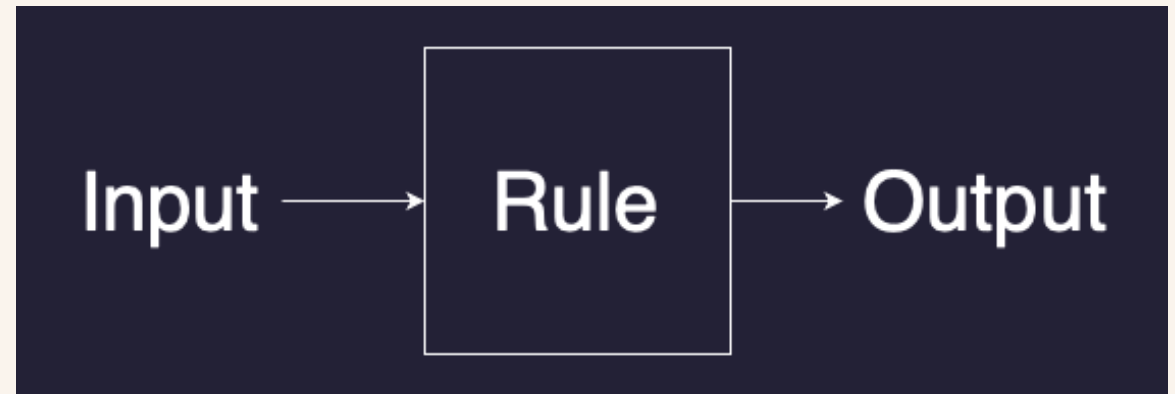
## ◀ Recap

```
if x < 10:  
    print('A')  
elif x >= 13:  
    print('B')  
elif x >= 20:  
    print('C')  
else:  
    print('D')
```

- `x = 1` ?
- `x = 13` ?
- `x = 10` ?
- `x = 11` ?
- `x = 20` ?

# Control

- conditionals: branching
- **loops**: repetition



# 1. Meow

meow  
meow  
meow

# Don't Repeat Yourself (DRY) principle

```
print("meow")  
print("meow")  
print("meow")
```

# Loop structures in Python

`while`

`for`

**while**: repeat while a condition is true

```
i = 3
while i > 0:
    print("meow")
    i = i - 1
```



## while using -= for assignment

```
i = 3
while i > 0:
    print("meow")
    i -= 1           # same as i = i - 1
```

# Assignment operators

- `=` : `x = 1`
- `+=` : `x += 1` (same as `x = x + 1`)
- `-=` : `x -= 1` (same as `x = x - 1`)
- `*=` : `x *= 2` (same as `x = x * 2`)
- `/=` : `x /= 2` (same as `x = x / 2`)
- ...

---

<https://python-reference.readthedocs.io/en/latest/docs/operators/>

## **for**: repeat over a sequence (list, string, ...)

```
for i in [0, 1, 2]:  
    print("meow")
```

```
for i in [0, 0, 0]:  
    print("meow")
```

```
for i in "abc":  
    print("meow")
```

**range( )** to generate a list of numbers

```
for i in range(3): # generates [0, 1, 2]
    print("meow")
```

## : throwaway variable

```
for _ in range(3):  
    print("meow")
```

## range(start, end)

```
for i in range(0, 3):    # same as range(3)
    print(i)
```

```
for i in range(start=5, end=9):
    print(i)
```



## 2. Printing even numbers between 1 and 20

- Use a `for` loop and the `range` function to iterate from 1 to 20 (inclusive).
- Inside the loop, use an `if` statement to check if the current number is even.
  - To check for evenness, use the modulo operator `%`.
- If the number is even, print it.

```
2
4
6
8
10
12
14
16
18
20
```

### 3. Interactive meow

```
Enter a positive number: -3
Enter a positive number: -1
Enter a positive number: 4
meow
meow
meow
meow
```

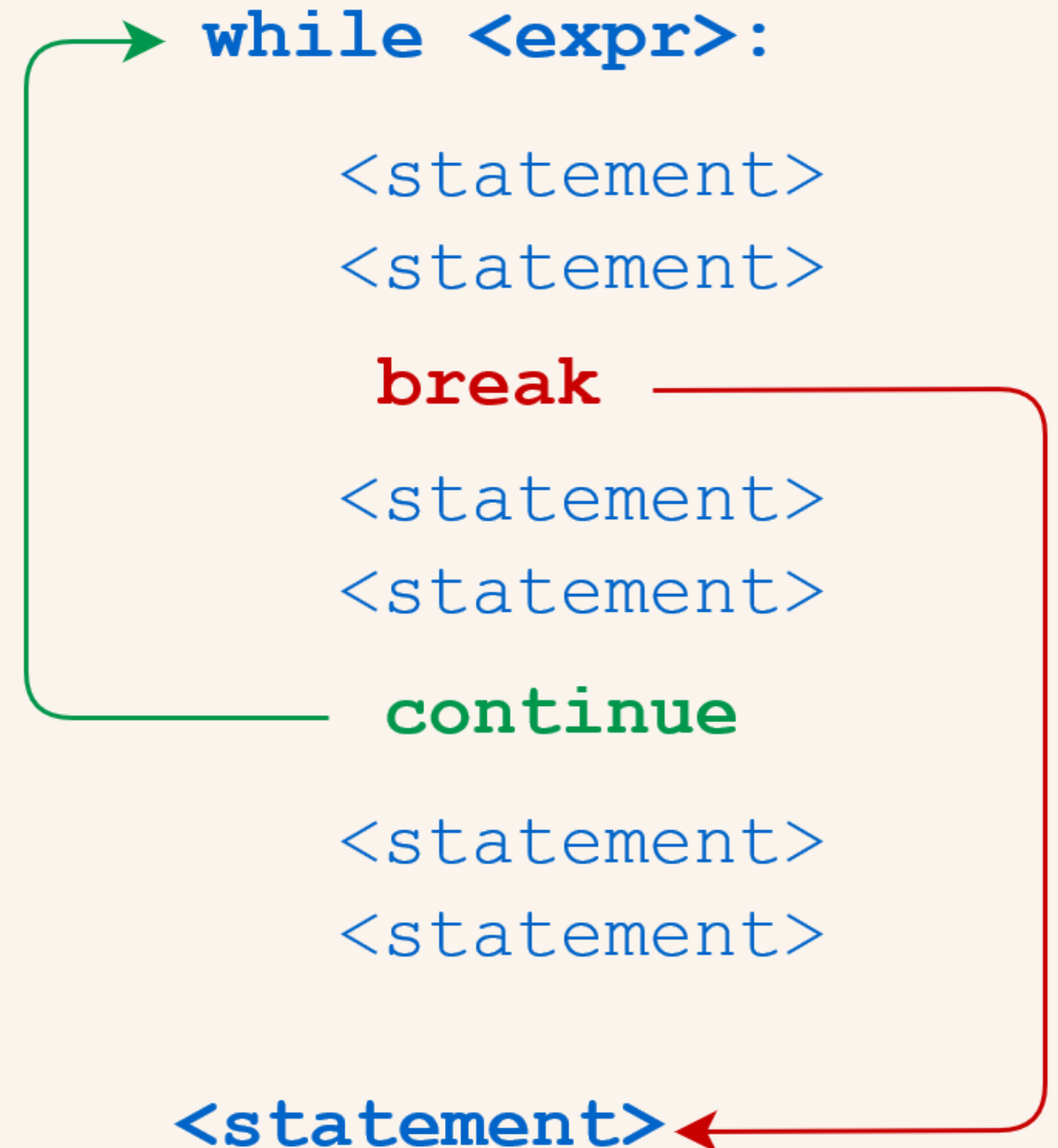


# Infinite loop to repeat forever

```
while True:                # while condition is always true
    print("meow")
```

## How to escape a loop?

- `continue`: continue to the next iteration
- `break`: break out of the loop



# How to escape a loop?

```
while True:                # infinite loop
    n = int(input("Enter a positive number: "))

    if n < 0:               # if n is negative
        continue          # continue to the next iteration
    else:                  # if n is non-negative
        break              # break out of the loop

for _ in range(n):
    print("meow")
```

# How to escape a loop?

```
while True: # infinite loop
    n = int(input("Enter a positive number: "))

    if n > 0: # if n is positive
        break # break out of the loop

for _ in range(n):
    print("meow")
```

# How to escape a loop using **return**

```
def main():
    # Ask the user to enter a positive number
    n = get_positive_number()
    # Print "meow" n times
    meow(n)

def get_positive_number():
    while True:
        n = int(input("Enter a positive number: "))
        if n > 0:
            return n # return the number

def meow(n):
    for _ in range(n):
        print("meow")

main()
```

# What will be printed? 🤔

```
for i in range(5):  
    if i == 2:  
        continue  
    print(i)
```

# What will be printed? 🤔

```
for i in range(5):  
    if i == 2:  
        break  
    print(i)
```

# What will be printed? 🤔

```
def mystery():  
    for i in range(5):  
        if i == 2:  
            return i  
        print(i)  
    return -1  
  
print(mystery())
```



# What will be printed? 🤔

```
for i in range(5):  
    if i % 2 == 0:  
        continue  
    if i > 2:  
        break  
    print(i)
```



## 4. Input Validation for Even Numbers

- Use a `while True` loop to prompt the user to enter a number until an even number is entered.
- Write a function `is_even(n)` that takes an integer `n` and returns `True` if `n` is even and `False` otherwise.
- Use an `if` statement to check whether the entered number is even.
  - If the number is even, return it and break out of the loop.
- Calculate the square of the returned even number and print it.

```
Enter an even number: 3
3 is not an even number. Try again.
Enter an even number: 5
5 is not an even number. Try again.
Enter an even number: 8
64
```