

Control

Recap

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

Recap

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

Control

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



1. Meow

Don't Repeat Yourself (DRY)

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

Loop

`while`

`for`

while: conditionally repeated

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


while using -= for assignment

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

Assignment operators

- `=`
- `+=`
- `-=`
- `*=`
- `/=`
- ...

<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()

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

for throwaway variable

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

range(start, end)

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

```
for i in range(5, 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

```
while True:  
    print("meow")
```

How to get out of a loop? **continue**, **break**

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

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

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


infinite loop continues anyway

```
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")
```

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

return to break out of a loop

```
def main():
    n = get_positive_number()
    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()
```



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
```