

```
# classwork Nov 3, 2019
```

```
# if else, elif
```

```
grade = 84
if grade > 80:
    print("It's a PASS")
else:
    print("It's a FAIL")
if grade >= 90:
    print("Your grade is A")
elif grade >= 80:
    print("Your grade is B")
elif grade >= 70:
    print("Your grade is C")
else:
    print("Your grade is F")
```

```
# prime numbers
```

```
for i in range(2, 101):
    notprime = 0
    for j in range(2, int(i/2)):
        if i%j == 0:
            notprime = 1
            print(f"{i} is not a prime")
            break
    if notprime == 0:
        print(f"{i} is a prime -----")
```

```
# list is a collection of objects
```

```
# list elements can be accessed by Index
```

```
# Lists have index values (locations?)
```

```
a = ["a", "b", "c", "d", "e", "f"]
```

```
# lists are ordered
```

```
a = [1, 2, 3]
b = [1, 3, 2]
print(a == b)
print(a[0] == b[0])
```

```

# list elements can be any type, same or different
b = [1, 3.14, "hello"]

# empty list
a = []
for i in range(10):
    a.append(i)
print(a)

# notice that string elements (letters)
# can be accessed by index as well
print(b[2])
print(b[2][4])

# negative index (starts with -1, the end)
print(b[-1])
print(b[-1][-1])

# this can be used to modify string values
b[1] = "this is a new value"
print(b)

# slicing
# (including:excluding)
print(b[1:3])

# reverse
print(b)
print(b[::-1])
print("Hello!"[::-1])

# in and not in operators
print(3.14 in b)
print(3.14 not in b)

c = ["ok", 999] + b
print(c)

d = b + [17]
# cant do d = b + 17

```

```

print(b)
print(f"length of b is {len(b)}")

# nested lists
nested = ["zero index", b, c]
print(nested)
print(nested[1][2][1])

print(c)
del c[3]
print(c)

c[2:2] = ["this is new", 2019]
print(c)

c[2] = []
print(c)

c[2:3] = []
print(c)

d1 = c[:]
d2 = c[:]

# difference between append and extend
d1.append(["new", "year", 2020])
d2.extend(["new", "year", 2020])
print(d1)
print(d2)

# insert!
c.insert(2, "this is new again")
print(c)

c.remove(2019)
print(c)

c.pop(1)
print(c)

```

