Lesson 3

Python Essentials

Overview

1. Conditional Statments
2. Error Handling
3. Functions
4. Lambda functions

Conditional Statements

```
country = 'CANADA'
if country == 'canada':
    print('Oh look a Canadian')
elif country == 'england':
    print('Oh look an english gentleman')
else:
    print('Not sure where you live')
```

Symbol	Operation
>	Greater than
<	Less than
>=	Greater than or equal to
<=	Less than or equal to
==	is equal to
!=	is not equal to
x in [a,b,c]	Does x match the value of a, b, or c

How you indent your code changes execution

String comparisons are case sensitive

Use string functions to make case insensitive comparisons .lower()

Complex conditions

```
if gpa >= .85:
    if lowest_grade >= .70:
        print('Well done')

if gpa >= .85 and lowest_grade >= .70:
    print('Well done')
```

First	Second	And	Or
TRUE	TRUE	TRUE	TRUE
TRUE	FALSE	FALSE	TRUE
FALSE	TRUE	FALSE	TRUE
FALSE	FALSE	FALSE	FALSE

Requirements for honour roll

Minimum 85% grade point average

Lowest grade is at least 70%

Handling runtime error

- Recover from error state
- Logging
- Graceful exit

```
try:
    print(x / y)
except ZeroDivisionError as e:
    # Optionally, log e somewhere
    print('Sorry, something went wrong')
except:
    print('Something really went wrong')
finally:
    print('This always runs on success or failure')
```

Notes:

- handle from more specific to more generic
- When to use:
  - User input

  - Accessing an external system
  - REST call
  - File system

Functions

- Recover from error state
- Logging
- Graceful exit

```
def get_initial(name):
    initial = name[0:1].upper()
    return initial

first_name = input('Enter your first name: ')
first_name_initial = get_initial(first_name)
last_name = input('Enter your last name: ')
last_name_initial = get_initial(last_name)
```

```
Enter your first name: susan
Enter your last name: ibach
Your initials are: SI
```

Lambda Functions

- Inline function
- Anonymous, may not have a name
- Frequently used with higher-order functions which take functions as argumetns

```
(lambda x: x + 1)(2)
```

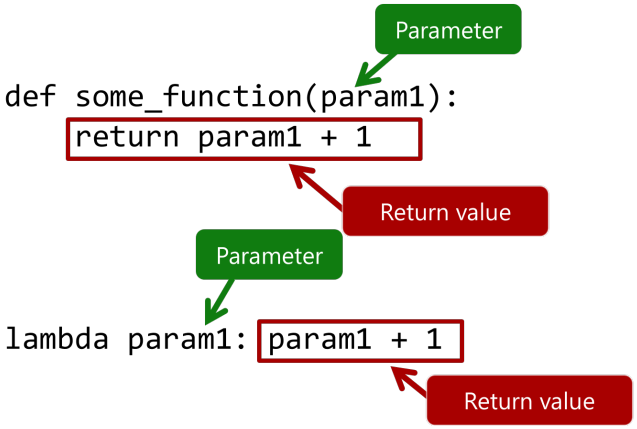

Figure 1: image