Lessons 1-2

Subjects:

1. ### Setting Up Codding Environment

   - [ ] **Downloading And Setting Up Python**
   - [ ] **Downloading And Setting Up VSCode**

2. ### Programming Knowledge

  - [ ] Explaining About The Python Programming Language And It's Uses
  - [ ] General Knowledge

Lessons 3-5

Subjects:

1. ## Basic Functions

```
print()         # the print() function
input()         # the input() function
```

2. ## Comments

```
# this is a comment indicated by the *#*
```

3. ## Variables

  - [ ] **int** - `Intgers (Whole Numbers)`
  - [ ] **float** - `Floating Point Number (Decimal Numbers)`
  - [ ] **str** - `String (Words Or Characters)`
  - [ ] **bool** - `Boolean (True Or False)`
  - [ ] **Changing The Type Of An Integer** - `str() int() bool() eval()`

4. ## Math

  - [ ] **Addition** - `Adding Numbers`
  - [ ] **Subtraction** - `Subtracting Numbers`
  - [ ] **Multiplication** - `Multiplyng Numbers`
  - [ ] **Division** - `Taking The Square Root Of A Number`
  - [ ] **Exponentiation** - `Taking The Square Root Of A Number`

- [ ] **Root Square** - `Taking The Square Root Of A Number`
- [ ] **Quick Addition**
- [ ] **Quick Subtraction**
- [ ] **Quick Multiplication**

Lessons 6-8

Subjects:

1. String Manipulation

- [ ] **Getting A Character** - `name[x]`
- [ ] **Getting A Sequence Of Characters** - `name[start:end]`
- [ ] **Getting A Sequence Of Characters With Jump Interval** - `name[start:end:jump]`
- [ ] **Reversed String Manipulation** - `name[start:end:jump]` `start,end,jump Can Be Negative Numbers`

2. If Statements

- [ ] **Fundamentals Of If Statements**
- [ ] **The Usages Of If Statements**
- [ ] **If Elif Else**
- [ ] **and or**
- [ ] **Indentation**

Lessons 9-12

Subjects:

1. Lists

- [ ] **List Declaration** - `name = [x]`

- ☐ **List Usages**
- ☐ **Joining A List Into A String** - `"filler".join(list)`
- ☐ **Get An Item** - `name[x]`
- ☐ **Get A Sequence Of Items** - `name[start:end]`
- ☐ **Get A Sequence Of Items With Jump Interval** - `name[start:end:jump]`
- ☐ **Reversed** - `name[start:end:jump] start,end,jump Can Be Negative`
- ☐ **Change A Item** - `name[y] = x`
- ☐ **Reverse A List** - `reversed(list), list.reverse(), Tip: can also be done by list[::-1]`
- ☐ **Add An Item To A List** - `name.append(x)`
- ☐ **Remove An Item From A List** - `name.remove(x)`
- ☐ **Get The Amount Of Items In A List** - `len(name)`

2. Tuples

- ☐ **Tuple Declaration** - `name = [x]`
- ☐ **Diffreces Between Tuples And Lists**
- ☐ **Tuples Usages**
- ☐ **Get An Item** - `name[x]`
- ☐ **Get A Sequence Of Items** - `name[start:end]`
- ☐ **Get A Sequence Of Items With Jump Interval** - `name[start:end:jump]`
- ☐ **Reversed** - `name[start:end:jump] start,end,jump Can Be Negative`
- ☐ **Get The Amount Of Items In A Tuple** - `len(name)`

3. Sets

- ☐ **Set Declaration** - `name = {x}`
- ☐ **Diffreces Between Sets And Lists** - `Sets Can't Have Multiple Of The Same Item While Lists Can`
- ☐ **Sets Usages**
- ☐ **Get The Amount Of Items In A Set** - `len(name)`

4. Dictionaries

- ☐ **Dictionaries Declaration** - `name = {key:value, key:value,...}`
- ☐ **Dicts Usages**
- ☐ **Get A Value By The Key** - `name[key]`
- ☐ **Get All Of The key:value Pairs In A Dict** - `name.items()`
- ☐ **Get All Of The Keys In A Dict** - `name.keys()`
- ☐ **Get The Amount Of Items In A Dict** - `len(name)`

5. Functions

- ☐ **Functions Declaration** - `def name(params)`
- ☐ **Usages Of A Function**
- ☐ **Calling A Function** - `name(arguments)`
- ☐ **What Are Paramaters And Arguments**
- ☐ ***args and **kwargs** - `passing an unkwon amout of arguments or key:value pairs`
- ☐ **Whats Is return** - `At The End Of A Function We Ussually Return A Value That Can Be Stored`

6. For Loops

- ☐ **Syntax Of For Loops** - `for variable_name(We Usually Use The Letter i) in variable_num2: (Go A Line Down)`
- ☐ **Usages Of A For Loop**
- ☐ **break And continue** - `break Stops The Loop And continue Skips To The Next Interation Of The Loop`
- ☐ **Indentation In For Loops**
- ☐ **for else** - `If The Loop Manages To Finish Without Breaking The Code In The else Part Will Be Ran`