User Manual

Prepared by: Fatemeh Ebrahimi

Getting started

- ### Choosing File

  Once you open the application you must open a file which your code will be written in. It can be done by pushing **'Select New File'** button and creating your file.
  If you want to open your file separately in application, you can push **'Open Current File'** button.

- ### Speaking Your Code

  By pushing **'Start Listening…'** button you can start speaking and the generated code will be shown in the box in the right side.

  You must notice after pushing the button, you must start speaking after the message **'Speak now'** is shown.

How to Speak Your Code

Note: Name of variables and functions will be generated in `snake_case` format, based on the PEP-8 convention.

In this application you can use these operations:

- Defining functions
- Return in a function
- Calling a function
- Defining if/else/elif condition
- Variable declarations
  - for integer, float, string, list, dictionary, dynamic variables
  - for math operations

- ### Defining functions

  In order to define any function in python, you have to say the following format:

  ```
  define function {name_of_the_function} parameters
  {name_of_the_parameters} end of parameters
  ```

  and also, between every parameter, the user has to say **`next`.**

  for example:

  ```
  define function hello world parameters name end of parameters
  ```

and the code which will be generated is:

```
def hello_world(name):
```

After that, the body of the function should be defined. Finally, when the function is finished, the user has to say `**end of function**` **.**

- ## Return in a Function

When the function is finished, just before `**end of function**` keyword, you can return a static or dynamic variable with the command below:
If you want to return static variable, you have to say:

```
return {type_of_variable} {value_of_that}
```

for example:

```
return string hello world
```

and the output will be:

```
return 'hello world'
```

and, finally, if you want to return dynamic variable, you have to say:

```
return variable {name_of_variable}
```

for example:

```
 return variable hello world
```

and the output will be:

```
 return hello_world
```

- ## Calling a Function

for function calls, you have to follow the format below:

```
function call {name_of_the_function} parameters {type_of_variable}
{name_of_parameter} end of parameters
```

and also between every parameter, you have to say `**next**`.
for example:

```
function call hello world parameters variable name end of parameters
```

and the code which will be generated is:

```
hello_world(name)
```

another example:

```
function call hello world parameters string name end of parameters
```

and the code which will be generated is:

```
hello_world('name')
```

- **Defining If/Else/Elif Condition**

In order to define any ELSE condition in python, user has to say the following format:

```
else condition
```

In order to define any IF/ELIF condition in python, user has to say the following format:

```
{If/else if} condition {type_of_variable} {name_of_variable} is
{type_of_comparison} {type_of_variable} {name_of_variable}
```

- **'type_of_the_variable'** can be:
  - `variable` for dynamic variables.
  - `integer`
  - `float`
  - `string`

- **' type_of_the_compare'** can be:
  - `equal to`
  - `not equal to`
  - `less than`
  - `greater than`
  - `less than or equal to`
  - `greater than or equal to`

for example:

```
if condition variable temp number is equal to integer 22
```

and the code which will be generated is:

```
if temp_number == 22:
```

- **Variable Declaration**
  - **for integer, float, string, list, dictionary, dynamic variables**

In order to declare any variable in python, you have to say the following format:

```
variable {name_of_variable} is {type_of_variable} {value_of_variable}
```

For the type of the variable, 6 options are available:

- **Integer**
- **Float**
- **String**
- **List**
- **Dictionary**
- **Variable (for dynamic variables)**

And the format of the value of the variable is dependent on the type of the variable:

- **Integer Number**

  You have to say the number, exactly.

  *-Example***:**

  Input:

  ```
  variable my integer number is integer 22
  ```

  Output:

  ```
  my_integer_number = 22
  ```

- **Floating Number**

  At first, you have to say the integer part of the number, and after that, for separating the integer part from the decimal part, the word '**point'** has to be said, and finally the decimal part.

  -Example:

  Input:

  ```
  variable my floating number is float 7 point 56
  ```

  Output:

  ```
  my_floating_number = 7.56
  ```

- **Strings**

  The exact string has to be said, word by word. If there is a number in the string, the number will be interpreted as digits, but if you want to use letters instead of digits, the word '**letters'** has to be said.

  -*Examples:*

  Input:

  ```
  variable my first string is string my name is sina
  ```

  Output:

  ```
  my_first_string = 'my name is sina'
  ```

  Input:

  ```
  variable my second string is string consider the number 42
  ```

  Output:

  ```
  my_second_string = 'consider the number 42'
  ```

Input:

```
variable my third string is string consider the number letters 42
```

Output:

```
my_third_string = 'consider the number forty-two'
```

- **Lists**

  In this case, you have to say the type of the variable in the list and value of that and after that, if you want to add other variables, the word '**next'** has to be said, and at last the words '**end of list**' has to be said.

  -Example:

  Input:

  ```
  variable my list is list integer 5 next float 4 point 6 next string sina end of list
  ```

  Output:

  ```
  my_list = [5, 4.6, 'sina']
  ```

- **Dictionaries**

  You have to say the key and the value, respectively, with the type of them, if you want to add other variables, the word '**then'** has to be said, and at last the words '**end of dictionary**' has to be said.

  -Example:

  Input:

  ```
  variable my dictionary is dictionary string age integer 21 then float 3 point 6 list string sina next integer 5 end of list end of dictionary
  ```

  Output:

  ```
  my_dictionary = {'age': 21, 3.6: ['sina', 5]}
  ```

- **Variables**

  If you want to set a variable equal to another variable, you have to say the exact name of the second variable.

  -Example:

  Input:

  ```
  variable my first variable is variable my second variable
  ```

  Output:

  ```
  my_first_variable = my_second_variable
  ```

- **For Math Operations**

  When you want to declare a variable using math operations such as add, subtract, multiply and divide you have to say it in the following structure:

  ```
  variable {name_of_variable} is operation {type_of_operation}
  {type_of_first_variable} {name/value_of_first_variable}
  {proper_conjunction}
  {type_of_second_variable} {name/value_of_second_variable}
  ```

Operation types and their conjunction:

- **Add To**
- **Subtract From**
- **Multiply By**
- **Divide By**

Variable types:

- **Integer**
- **Float**
- **String**
- **List**
- **Variable (for dynamic variables)**

Examples:

- **Add**

  Input:

  ```
  variable first var is operation add string my string to integer 2
  ```

  output:

  ```
  first_var = "my string" + "2"
  ```

- **Subtract**

  Input:

  ```
  variable second var is operation multiply float 2 point 3 by
  integer 2
  ```

  output:

  ```
  second_var = 2.3 * 2
  ```

- **Multiply**

  Input:

  ```
  variable third var is operation subtract variable x from integer 4
  ```

  output:

  ```
  third_var = 4 - x
  ```

- **Divide**

  Input:

  ```
  variable fourth var is operation divide variable my var by
  variable y
  ```

  output:

  ```
  fourth_var = my_var / y
  ```

How to Remove Your Code

If you want to delete a specific code line you must say the following format:

```
Remove line {line_number}
```

How to Change the Line

If you want to change the line the specific line number, command below can be used:

```
Go to line {line_number/Start/End}
```