Hands-On GenAI: LLMs, RAGs, and Agentic Systems for Beginners

-Adya Bhat and Tejas Venugopalan

Agenda

What does an Agent look like? (contd.)

A quick revision of Python

What does an Agentic Application look like?

A fascinating video of AI agents in action

1000 AI NPCs simulate a CIVILIZATION in Minecraft

What is-

1. a program?
2. the web application/ agentic system built of?
3. a programming language?
4. Python, the programming language? Why Python?

Python- a high level language

cpython / Objects / **listobject.c**

Code | Blame | 4255 lines (3823 loc) · 126 KB

```c
#ifdef Py_GIL_DISABLED
typedef struct {
    Py_ssize_t allocated;
    PyObject *ob_item[];
} _PyListArray;

static _PyListArray *
list_allocate_array(size_t capacity)
{
    if (capacity > PY_SSIZE_T_MAX/sizeof(PyObject*) - 1) {
        return NULL;
    }
    _PyListArray *array = PyMem_Malloc(sizeof(_PyListArray) + capacity * sizeof(PyObject *));
    if (array == NULL) {
        return NULL;
    }
    array->allocated = capacity;
    return array;
}

static Py_ssize_t
list_capacity(PyObject **items)
{
    _PyListArray *array = _Py_CONTAINER_OF(items, _PyListArray, ob_item);
    return array->allocated;
}
#endif
```

Source: https://github.com/python/cpython/blob/main/Objects/listobject.c

CPython is an implementation of Python in C programming language, a widely used implementation.
(https://github.com/python/cpython)

Program- needs an input, has an output

1. input: from the keyboard
    - `input()`
2. output: on the console/ into a file of your choice
    - `print()`

Program- has some instructions, processes some data

data:

- literals, variables
- data structures (in-built in Python):
  - list: `[1, 2, 3, 3, 4, 5, "hello"]`
  - tuple: `(1, 2, 3, 3, 4, 5, "hello")`
  - dictionary:

    ```
    {"key1": "Student1", "key2": "Student2", "key3":
    "Student3"}
    ```

  - set: `{1, 2, 3, 4, 5}`

Program- has some instructions, processes some data

data structures:

	Ordered?	Subscriptable?	Mutable?	Unique?
Lists	Yes	Yes (index)	Yes	No (Duplicate elements allowed)
Tuples	Yes	Yes (index)	No	No (Duplicate elements allowed)
Dictionaries	No	Yes (key)	Yes	Unique Keys
Sets	No	No	Yes	Unique Elements

Let’s look at some code examples!

Program- has some instructions, processes some data

instructions:

- conditions:
  - `if-else`
  - `if-elif-else`
- loops:
  - `for`
  - `while`
- functions:
  -
```
def find_difference(param1, param2):
    return param1-param2

difference = find_difference(2, 3)
```

analogous to math functions!

Let’s look at some code examples!

Modules and Packages

Module: single file, containing reusable code, can use for built-in functions

- example: random, math
- math module: https://github.com/python/cpython/blob/main/Modules/mathmodule.c

Package: directory of modules

- example: numpy
- numpy: https://github.com/numpy/numpy/blob/main/numpy/_core/numeric.py

Conclusion

What we have covered today:

1. input- output
2. literals and variables
3. built-in data structures
4. conditions
5. loops
6. functions
7. modules and packages

Thank You!