# notes

September 11, 2023

## 1 Comments

```
[1]: # This is a comment
     '''
     This is also a comment
     '''

     print('Hello World')
```

```
Hello World
```

## 2 Multiple statements in single line

```
[2]: a = 3; print(a)
```

```
3
```

## 3 Multiline statements

```
[3]: a = 1 + 3 + 4 + \
         5 + 6 + 7
     print(a)
```

```
26
```

## 4 Multiple variable assignmeents

```
[4]: a,b = 4,5
     print(a,b)
```

```
4 5
```

# 5  help() & __doc__

```python
[5]: def my_func():
         '''This is a doc-string'''
         pass

     help(my_func)
     print('-'*100)
     print(my_func.__doc__)
```

```
Help on function my_func in module __main__:

my_func()
    This is a doc-string


------------------------------------------------------------------------------
--------------------
This is a doc-string
```

# 6  print() and input()

```python
[1]: print?
```

```
Signature: print(*args, sep=' ', end='\n', file=None, flush=False)
Docstring:
Prints the values to a stream, or to sys.stdout by default.

sep
  string inserted between values, default a space.
end
  string appended after the last value, default a newline.
file
  a file-like object (stream); defaults to the current sys.stdout.
flush
  whether to forcibly flush the stream.
Type:      builtin_function_or_method
```

```python
[2]: input?
```

```
Signature: input(prompt='')
Docstring:
Forward raw_input to frontends

Raises
------
StdinNotImplementedError if active frontend doesn't support stdin.
File:      /usr/local/lib/python3.11/site-packages/ipykernel/kernelbase.py
Type:      method
```

```
[6]: %%writefile my_module.py
     def greet():
         print('Hello World!')
```

Overwriting my_module.py