

1.8 Builtins

```
In [1]: import builtins
```

We can see what all builtins does Python provide.

For our sake, we are traversing the complete list and printing the number and builtin attribute.

The function we are using to traverse in `dir(builtins)` and get index and builtin attribute is `enumerate` which is also a builtin 😊

```
In [2]: for index, builtin_attribute in enumerate(dir(builtins)):
        print(f"{index}) {builtin_attribute}")
```

```
0) ArithmeticError
1) AssertionError
2) AttributeError
3) BaseException
4) BlockingIOError
5) BrokenPipeError
6) BufferError
7) BytesWarning
8) ChildProcessError
9) ConnectionAbortedError
10) ConnectionError
11) ConnectionRefusedError
12) ConnectionResetError
13) DeprecationWarning
14) EOFError
15) Ellipsis
16) EnvironmentError
17) Exception
18) False
19) FileNotFoundError
20) FileNotFoundError
21) FloatingPointError
22) FutureWarning
23) GeneratorExit
24) IOError
25) ImportError
26) ImportWarning
27) IndentationError
28) IndexError
29) InterruptedError
30) IsADirectoryError
31) KeyError
32) KeyboardInterrupt
33) LookupError
34) MemoryError
35) ModuleNotFoundError
36) NameError
37) None
38) NotADirectoryError
39) NotImplemented
40) NotImplementedError
41) OSError
42) OverflowError
43) PendingDeprecationWarning
44) PermissionError
45) ProcessLookupError
46) RecursionError
47) ReferenceError
```

48) ResourceWarning
49) RuntimeError
50) RuntimeWarning
51) StopAsyncIteration
52) StopIteration
53) SyntaxError
54) SyntaxWarning
55) SystemError
56) SystemExit
57) TabError
58) TimeoutError
59) True
60) TypeError
61) UnboundLocalError
62) UnicodeDecodeError
63) UnicodeEncodeError
64) UnicodeError
65) UnicodeTranslateError
66) UnicodeWarning
67) UserWarning
68) ValueError
69) Warning
70) ZeroDivisionError
71) __IPYTHON__
72) __build_class__
73) __debug__
74) __doc__
75) __import__
76) __loader__
77) __name__
78) __package__
79) __spec__
80) abs
81) all
82) any
83) ascii
84) bin
85) bool
86) breakpoint
87) bytearray
88) bytes
89) callable
90) chr
91) classmethod
92) compile
93) complex
94) copyright
95) credits
96) delattr
97) dict
98) dir
99) display
100) divmod
101) enumerate
102) eval
103) exec
104) filter
105) float
106) format
107) frozenset
108) get_ipython
109) getattr
110) globals
111) hasattr
112) hash
113) help
114) hex
115) id
116) input

```
117) int
118) isinstance
119) issubclass
120) iter
121) len
122) license
123) list
124) locals
125) map
126) max
127) memoryview
128) min
129) next
130) object
131) oct
132) open
133) ord
134) pow
135) print
136) property
137) range
138) repr
139) reversed
140) round
141) set
142) setattr
143) slice
144) sorted
145) staticmethod
146) str
147) sum
148) super
149) tuple
150) type
151) vars
152) zip
```

There's a difference between **Keywords** and **Builtins** 😞. We can't assign a new object to the Keywords, if we try to do, we would be seeing an exception raised 🛑. But coming to builtins, we can assign any object to the builtin names, and Python won't have any issues, but it's not a good practice to do so 😇