Python Keywords and More

Python Keywords

There are specific words that have a pre-existing meaning to the Python interpreter, these are referred to as *keywords* or *reserved words*, and cannot be used as variable names. We have seen that `True` and `False` are reserved, and in total there are 35 keywords in the current version of Python (which we'll learn a lot more about in subsequent pages), as follows:

and	**continue**	**finally**	**is**	**raise**
as	**def**	**for**	**lambda**	**return**
assert	**del**	**from**	**None**	**True**
async	**elif**	**global**	**nonlocal**	**try**
await	**else**	**if**	**not**	**while**
break	**except**	**import**	**or**	**with**
class	**False**	**in**	**pass**	**yield**

Python Built-In Functions

Other words that we have to avoid using when we are creating variables are the names of built-in functions. The `print` command is an example of a built-in function, so when we do `print(“Hello, World!”),` we call the function name – `print` – we follow it with brackets, that encloses some content, in this case it's the string `“Hello, World!”` but we've seen it could also be a number, a character, or even a Boolean. And we know that a built-in function does something, so in this case, it writes the string you have enclosed in brackets onto the screen. All functions work the same way, they take in value and perform a particular operation:

abs	**compile**	**format**	**isinstance**	**object**	**set**
all	**complex**	**frozenset**	**issubclass**	**oct**	**setattr**
any	**delattr**	**getattr**	**iter**	**open**	**slice**
ascii	**dict**	**globals**	**len**	**ord**	**sorted**
bin	**dir**	**hasattr**	**list**	**pow**	**staticmethod**
bool	**divmod**	**hash**	**locals**	**print**	**str**
bytearray	**enumerate**	**help**	**map**	**property**	**sum**
bytes	**eval**	**hex**	**max**	**range**	**super**
callable	**exec**	**id**	**memoryview**	**repr**	**tuple**
chr	**filter**	**input**	**min**	**reversed**	**type**
classmethod	**float**	**int**	**next**	**round**	**vars**