Python Most important Built-in-Modules list

Module Name	Purpose	Key Methods
math	Mathematical functions and constants	`math.sqrt(x)`, `math.sin(x)`, `math.cos(x)`, `math.pi`, `math.e`
random	Generate random numbers and make random choices	`random.random()`, `random.randint(a, b)`, `random.choice(seq)`, `random.shuffle(seq)`
os	Interaction with the operating system	`os.getcwd()`, `os.listdir(path)`, `os.path.join(path, *paths)`, `os.path.exists(path)`
sys	Access to system-specific parameters and functions	`sys.argv`, `sys.platform`, `sys.exit(status)`
datetime	Work with dates, times, and timedeltas	`datetime.datetime.now()`, `datetime.date(year, month, day)`, `datetime.timedelta(days, seconds, microseconds, milliseconds, minutes, hours, weeks)`
collections	Specialized container datatypes	`collections.deque(iterable)`, `collections.Counter(iterable)`, `collections.defaultdict(default_factory)`
json	Encode and decode JSON	`json.dumps(obj)`, `json.loads(json_str)`, `json.dump(obj, file_obj)`, `json.load(file_obj)`
re	Regular expression operations	`re.search(pattern, string)`, `re.match(pattern, string)`, `re.findall(pattern, string)`, `re.sub(pattern, repl, string)`
urllib	Work with URLs and make HTTP requests	`urllib.request.urlopen(url)`, `urllib.request.urlretrieve(url, filename)`, `urllib.parse.urlencode(params)`
argparse	Parse command-line arguments and options	`argparse.ArgumentParser()`, `.add_argument()`, `.parse_args()`
csv	Read and write CSV files	`csv.reader(file_obj)`, `csv.writer(file_obj)`, `csv.DictReader(file_obj)`, `csv.DictWriter(file_obj, fieldnames)`
pickle	Object serialization and deserialization	`pickle.dump(obj, file_obj)`, `pickle.load(file_obj)`
gzip	Work with gzip-compressed files	`gzip.open(filename, mode)`, `gzip.compress(data)`
sqlite3	Interaction with SQLite databases	`sqlite3.connect(database)`, `connection.execute(sql)`, `connection.commit()`, `connection.close()`
time	Time-related functions	`time.time()`, `time.sleep(seconds)`, `time.localtime()`, `time.strftime(format)`
itertools	Tools for efficient iteration	`itertools.product(iterables, repeat)`, `itertools.permutations(iterable, r)`, `itertools.combinations(iterable, r)`
functools	Higher-order functions and callable objects	`functools.partial(func, *args, **keywords)`, `functools.reduce(func, iterable[, initializer])`
os.path	Functions for path manipulation	`os.path.join(path, *paths)`, `os.path.dirname(path)`, `os.path.basename(path)`, `os.path.exists(path)`