Net ID:_____ **Name:**_____

Reference Material

Built-in

```
abs
bool
chr
dict
enumerate (sequence)
filter    (function,
iterable)
float
format
input
int
len
list
map     (function, iterable)
max     (iterable, key=func)
min     (iterable, key=func)
open    (filename, mode)
ord
pow
print
range
round
set
sorted (iterable, key=func)
str
sum     (iterable)
tuple
type
zip     (*iterables)
```

String Methods (most can also be used on str accessor in Series)

```
capitalize
count
endswith
find
format
index
isalnum
isalpha
isdecimal
isdigit
islower
isnumeric
isprintable
isspace
istitle
isupper

join (list) – concatenates strings in
list using string that join is called on

lower
replace

split – default delimiter is any
whitespace (if no argument)

startswith

strip – remove leading and trailing
whitespace

title
upper
```

Math Module

```
acos
acosh
asin
asinh
atan
atan2
atanh
ceil
cos
cosh
degrees
floor
log
log10
log2
pi
pow
radians
sin
sinh
sqrt
tan
tanh
```

List Methods

```
append count
extend index
insert pop
remove reverse
sort
```

File Object Methods

```
read
readline
readlines
write
```

Regular Expression Module

```
find
search
findall
```

json Module

```
dumps
loads
```

csv Module Functions

```
DictReader
(file obj)

reader
(file object)
```

Dictionary Methods

```
get
items
keys
pop
popitem
update
values
```

random Module

```
randint
choice
shuffle
```

SQL Reference

STATEMENTS	CLAUSES	AGGREGATE	OPERATORS	TYPES
INSERT	FROM	avg(numeric/decimal, any int) returns numeric	||	smallint
SELECT	WHERE		+	integer
UPDATE	GROUP BY	count(any type) returns bigint	-	bigint
DELETE	HAVING		*	decimal
CREATE DATABASE	ORDER BY	max(any numeric, any string, any datetime) returns same as argument	/	numeric
CREATE TABLE	LIMIT		%	real
ALTER TABLE		min(any numeric, any string, any datetime) returns same as argument	>	double
			<	serial
			<=	bigserial
			>=	
		sum(numeric/decimal, any int) returns numeric type based on argument	=	money
			<>	
			IS	text
			IS NOT	varchar(n)
				timestamp
				timestamptz
				date

DataFrame Properties

at Access a single value for a row/column label pair.
axes Return a list representing the axes of the DataFrame.
blocks (DEPRECATED) Internal property, property synonym for as_blocks().
columns The column labels of the DataFrame.
dtypes Return the dtypes in the DataFrame.
iloc Purely integer-location based indexing for selection by position.
index The index (row labels) of the DataFrame.
loc Access a group of rows and columns by label(s) or a boolean array.
ndim Return an int representing the number of axes / array dimensions.
shape Return a tuple representing the dimensionality of the DataFrame.
size Return an int representing the number of elements in this object.
values Return a Numpy representation of the DataFrame.

str Accessor Methods

capitalize(self)
cat(self[, others, sep, na_rep, join])
count(self, pat[, flags])
extract(self, pat[, flags, expand])
extractall(self, pat[, flags])
find(self, sub[, start, end])
findall(self, pat[, flags])
index(self, sub[, start, end])
len(self)
lower(self)
match(self, pat[, case, flags, na])
normalize(self, form)
pad(self, width[, side, fillchar])
partition(self[, sep, expand])
repeat(self, repeats)
replace(self, pat, repl[, n, ...])
slice(self[, start, stop, step])
slice_replace(self[, start, ...])
split(self[, pat, n, expand])
startswith(self, pat[, na])
strip(self[, to_strip])
title(self)
upper(self)
isalnum(self)
isalpha(self)
isdigit(self)
isspace(self)
islower(self)
isupper(self)
isnumeric(self)

DataFrame Methods

abs(self) Return a Series/DataFrame with absolute numeric value of each element.
any(self[, axis, bool_only, skipna, level]) Return whether any element is True, potentially over an axis.
append(self, other[, ignore_index, ...]) Append rows of other to the end of caller, returning a new object.
apply(self, func[, axis, broadcast, raw, ...]) Apply a function along an axis of the DataFrame.
applymap(self, func) Apply a function to a Dataframe elementwise.
astype(self, dtype[, copy, errors]) Cast a pandas object to a specified dtype dtype.
bool(self) Return the bool of a single element PandasObject.
copy(self[, deep]) Make a copy of this object's indices and data.
drop(self[, labels, axis, index, columns, ...]) Drop specified labels from rows or columns.
dropna(self[, axis, how, thresh, subset, ...]) Remove missing values.
groupby(self[, by, axis, level, as_index, ...]) Group DataFrame or Series using a mapper or by a Series of columns.
head(self[, n]) Return the first n rows.
info(self[, verbose, buf, max_cols, ...]) Print a concise summary of a DataFrame.
isnull(self) Detect missing values.
join(self, other[, on, how, lsuffix, ...]) Join columns of another DataFrame.
keys(self) Get the 'info axis' (see Indexing for more)
max(self[, axis, skipna, level, numeric_only]) Return the maximum of the values for the requested axis.
mean(self[, axis, skipna, level, numeric_only]) Return the mean of the values for the requested axis.
median(self[, axis, skipna, level, numeric_only]) Return the median of the values for the requested axis.
merge(self, right[, how, on, left_on, ...]) Merge DataFrame or named Series objects with a database-style join.
min(self[, axis, skipna, level, numeric_only]) Return the minimum of the values for the requested axis.
mod(self, other[, axis, level, fill_value]) Get Modulo of dataframe and other, element-wise (binary
notnull(self) Detect existing (non-missing) values.
reindex(self[, labels, index, columns, ...]) Conform DataFrame to new index with optional filling logic, rename(self[, mapper, index, columns, axis, ...]) Alter axes labels.
sample(self[, n, frac, replace, weights, ...]) Return a random sample of items from an axis of object.
sort_index(self[, axis, level, ascending, ...]) Sort object by labels (along an axis).
sort_values(self, by[, axis, ascending, ...]) Sort by the values along either axis.
sum(self[, axis, skipna, level, ...]) Return the sum of the values for the requested axis.
tail(self[, n]) Return the last n rows.
to_csv(self[, path_or_buf, sep, na_rep, ...]) Write object to a comma-separated values (csv) file.
to_dense(self) (DEPRECATED) Return dense representation of Series/DataFrame (as opposed to sparse).

Requests

```
res = requests.get(url)   # retrieve url
res.status_code           # 404 not found, 200 OK
res.content               # resource content (html, json,
                          #etc.)
```

BeautifulSoup

```
dom = BeautifulSoup(html) # parse html string
dom.select(selector)      # get list of elements  using css
                          # selector
ele.select(selector)      # use css selector w/ in element
ele.tag_name              # first tag_name in element
ele.get_text()            # get text within tags
```

CSS Selectors

```
tag
.class
#id
selector1 selector2 – any selector2 child of selector1
selector1 > selector2 – selector2 direct descendant of selector1
selector1, selector2 – either selector1 or selector2
```