```
KOLEKCJE

class list
class list(iterable)
class tuple
class tuple(iterable)
class dict(**kwarg)
class dict(mapping, **kwarg)
class dict(iterable, **kwarg)
vars()
vars(object)
class set
class set(iterable)
class frozenset(iterable=set())
class range(stop)
class range(start, stop, step=1)
class slice(stop)
class slice(start, stop, step=None)
```

```
len(s)

filter(function, iterable)
map(function, iterable, *iterables)
any(iterable)
all(iterable)

reversed(seq)
sorted(iterable, /, *, key=None,
reverse=False)
zip(*iterables, strict=False)

sum(iterable, /, start=0)
max(iterable, *, key=None)
max(iterable, *, default, key=None)
max(arg1, arg2, *args, key=None)
min(iterable, *, key=None)
min(iterable, *, default, key=None)
min(arg1, arg2, *args, key=None)
```

```
s[i] = x
s[i:j] = t
del s[i:j]
s[i:j:k] = t
del s[i:j:k]
s.append(x)
s.clear()
s.copy()
s.extend(t) or s += t
s *= n
s.insert(i, x)
s.pop() or s.pop(i)
s.remove(x)
s.reverse()
```

```
SET, FROZEN SET

len(s)
x in s
x not in s
isdisjoint(other)
issubset(other)
set <= other
set < other
issuperset(other)
set >= other
set > other
union(*others)
set | other | ...
intersection(*others)
set & other & ...
difference(*others)
set - other - ...
symmetric_difference(other)
set ^ other
copy()
```

```
SET

update(*others)
set |= other | ...
intersection_update(*others)
set &= other & ...
difference_update(*others)
set -= other | ...
symmetric_difference_update(other)
set ^= other
add(elem)
remove(elem)
discard(elem)
pop()
clear()
```

```
aiter(async_iterable)
awaitable anext(async_iterator)
awaitable anext(async_iterator, default)

enumerate(iterable, start=0)
iter(object)
iter(object, sentinel)
next(iterator)
next(iterator, default)
```

DICTIONARY
 	
list(d)
len(d)
d[key]
d[key] = value
del d[key]
key in d
key not in d
iter(d)
clear()
copy()
classmethod fromkeys(iterable[, value])
get(key[, default])
items()
keys()
pop(key[, default])
popitem()
reversed(d)
setdefault(key[, default])
update([other])
values()
d = {'a': 1}
d | other
d |= other
	DICTIONARY VIEW OBJECT
	
	len(dictview)
	iter(dictview)
	x in dictview
	reversed(dictview)
	dictview.mapping	

NAPISY
 	
class str(object='')
class str(object=b'', encoding='utf-8', errors='strict')
	str.center(width[, fillchar])
	str.find(sub[, start[, end]])
	str.index(sub[, start[, end]])
	str.rfind(sub[, start[, end]])
	str.rindex(sub[, start[, end]])
	str.count(sub[, start[, end]])
	str.replace(old, new[, count])
	str.translate(table)
	static str.maketrans(x[, y[, z]])
	str.expandtabs(tabsize=8)
	str.join(iterable)
	
	str.endswith(suffix[, start[, end]])
	str.startswith(prefix[, start[, end]])
		str.isalnum()
		str.isalpha()
		str.isascii()
		str.isdecimal()
		str.isdigit()
		str.isidentifier()
		str.islower()
		str.isnumeric()
		str.isprintable()
		str.isspace()
		str.istitle()
		str.isupper()

```
str.capitalize()
str.casefold()
str.lower()
str.upper()
str.swapcase()
str.title()
str.encode(encoding='utf-8',
errors='strict')

str.strip([chars])
str.lstrip([chars])
str.rstrip([chars])

str.removeprefix(prefix, /)
str.removesuffix(suffix, /)

str.ljust(width[, fillchar])
str.rjust(width[, fillchar])
```

```
str.split(sep=None, maxsplit=- 1)
str.rsplit(sep=None, maxsplit=- 1)
str.partition(sep)
str.rpartition(sep)
```

```
str.format(*args, **kwargs)
str.format_map(mapping)
str.zfill(width)
```

```
repr(object)
format(value, format_spec='')
print(*objects, sep=' ', end='\n', file=None, flush=False)

input()
input(prompt)

open(file, mode='r', buffering=- 1, encoding=None, errors=None, newline=None, closefd=True, opener=None)
```

```
eval(expression, globals=None, locals=None)

compile(source, filename, mode, flags=0, dont_inherit=False, optimize=- 1)
exec(object, globals=None, locals=None, /, *, closure=None)

help()
help(request)
breakpoint(*args, **kws)

class memoryview(object)
```

LICZBY
 	
class int(x=0)
class int(x, base=10)
class float(x=0.0)
class complex(real=0, imag=0)
class complex(string)
	divmod(a, b)
	pow(base, exp, mod=None)
	round(number, ndigits=None)
	
	int.as_integer_ratio()
	is_integer()
		ascii(object)
		bin(x)
		oct(x)
		hex(x)
		class bool(x=False)
		chr(i)
		ord(c)

x+y
x-y
x*y
x/y
x//y
x%y
x**y
+x
-x
	abs(x)
	int(x)
	float(x)
	
	complex(re,im)
	c.conjugate()
	divmod(x,y)
	pow(x,y)
	
	math.trunc(x)
	round(x[,n])
	math.floor(x)
	math.ceil(x)
		<
		>
		<=
		>=
		==
		!=
		is
		is not

KLASY i OBIEKTY
 	
class object
class property(fget=None, fset=None, fdel=None, doc=None)
class super
class super(type, object_or_type=None)
 	
@classmethod decorator
@staticmethod decorator
	setattr(object, name, value)
	delattr(object, name)
	getattr(object, name)
	getattr(object, name, default)
	hasattr(object, name)
	
	locals()
	globals()
	dir()
	dir(object)
	callable(object)
	hash(object)
		id(object)
		isinstance(object, classinfo)
		issubclass(class, classinfo)