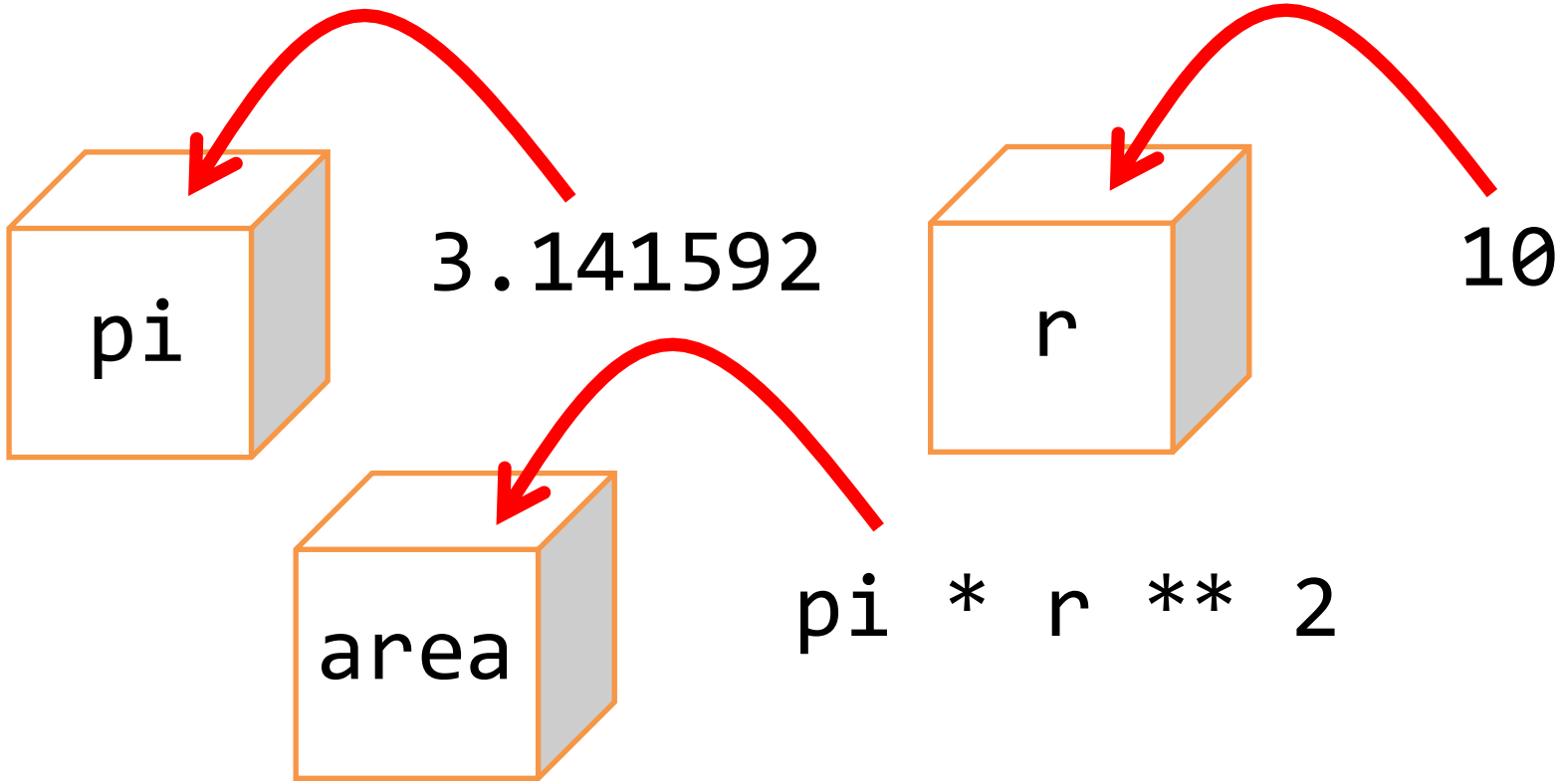
오늘의 강의 목표

- 변수(Variable)와 메모리(Memory)에 대한 이해
- 자료형(Data Type)에 대한 이해
- 형변환(Type Conversion)에 대한 이해

Variables

```
>>> pi = 3.141592
>>> r = 10
>>> area = pi * r ** 2
>>> area
314.1592
```



Creating Variables

- = (assignment) 연산자를 이용하여 변수 생성
- 생성된 변수는 변수명을 통해 접근


```
>>> my_age = 20              # create my_age variable
>>> print(my_age)
20
>>> your_age = my_age        # create your_age variable
>>> print(your_age)
20
```

```
>>> my_age = your_age = 20        # create two variables
```

Destroying Variables

- del 명령어를 이용하여 변수 삭제
- 삭제된 변수는 접근 불가

```
>>> year = 2015      # create year variable
>>> print(year)
2015
>>> del year         # destroy year variable
>>> print(year)
Traceback (most recent call last):
  File "<pyshell#14>", line 1, in <module>
    print(year)
NameError: name ‘year' is not defined
```

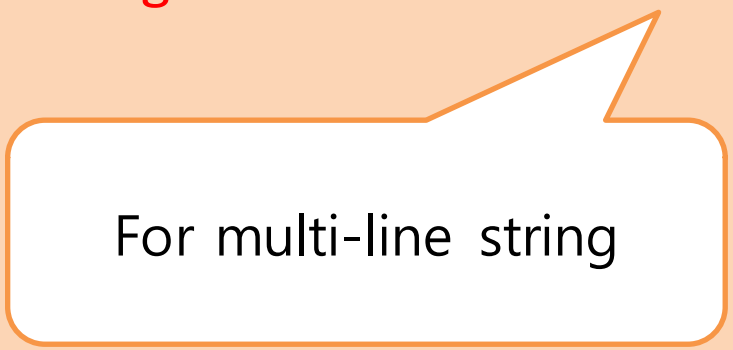
Working with Variables

```
>>> days = 3                    # create days
>>> hours = days * 24           # create hours with days
>>> hours
72
>>> days = days + 1             # update days
>>> days
4
>>> days += 1                   # a shortcut
>>> days
5
```

Creating String Variables

```
>>> my_name = "John"        # string variable with "
>>> print(my_name)
John
>>> your_name = 'Taylor'    # string variable with '
>>> print(your_name)
Taylor
>>> message = '''           # string variable with '''
Dear Professor,
My name is Jong-Chan.
Nice to see you.
'''
>>> print(message)

Dear Professor,
My name is Jong-Chan.
Nice to see you.
```



Deleting String Variables

```
>>> my_name = "John"
>>> print(my_name)
John
>>> del my_name
>>> print(my_name)
Traceback (most recent call last):
  File "<pyshell#111>", line 1, in <module>
    print(my_name)
NameError: name 'my_name' is not defined
```

Use del command

Variables and Memory

- 변수(Variable): 자료를 저장하기 위한 저장 장소
- 각 변수는 Memory의 일부 장소를 사용

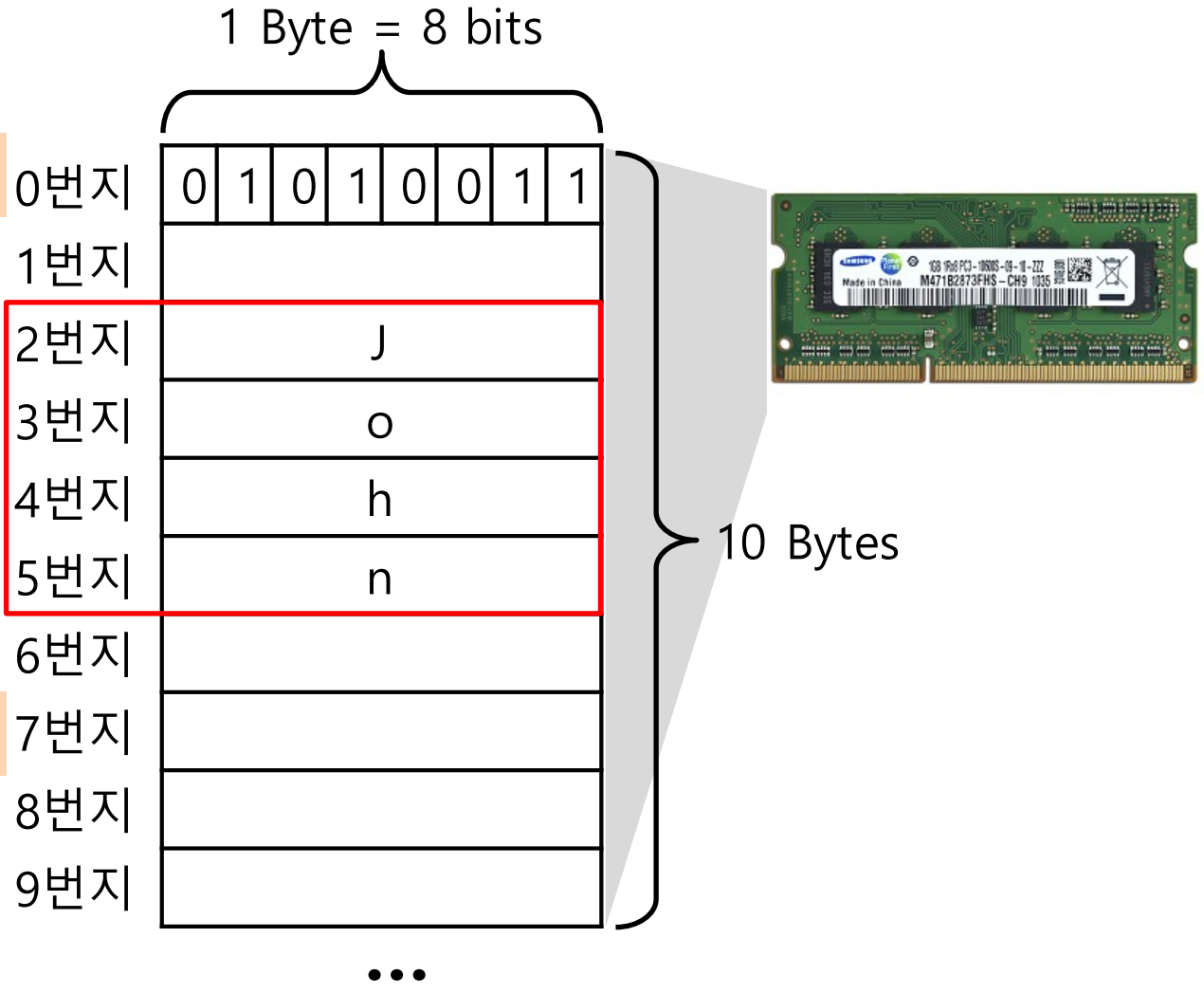
변수 생성

```
>>> my_name = "John"
```

my_name 변수에
할당된 메모리 영역

변수 삭제

```
>>> del my_name
```



Variables and Memory

- 각 변수가 차지하는 사이즈
  - sys.getsizeof()

```
>>> import sys
>>> a = 10
>>> sys.getsizeof(a)
28     # 28 Bytes for variable a
>>> b = "John"
>>> sys.getsizeof(b)
53     # 53 Bytes for variable b
>>> b = "John is working in the backyard"
>>> sys.getsizeof(b)
80     # Variable b grows to 80 Bytes
```

Python Data Types

- Numbers
  - 숫자 ex) 10, -78, 080, 0x80, 3.14, 1 + 3j
- String
  - 문자열 ex) "John", 'Hello World'
- List
  - 목록 ex) [1, 2, 3, 4], ['a', 'b', 'c'], [1, 'a', 'b', 2]
- Tuple
  - 조 ex) (1, 2), ('a', 'b', 'c')
- Dictionary
  - 사전 ex) {1:"John", 2:"Taylor"}

Python Number Types

- int
  - 정수
- float
  - 실수
- complex
  - 복소수

Python Number Types

```
>>> a = 10
>>> b = 10.0
>>> c = 10 + 10j
>>> type(a)
<class 'int'>
>>> type(b)
<class 'float'>
>>> type(c)
<class 'complex'>
```

type() 명령 이용하여 자료형 확인

Python Strings

```
>>> s = 'Hello World'
>>> type(s)
<class 'str'>
>>> len(s)          # string length
11
>>> print(s.upper())      # upper case
HELLO WORLD
>>> print(s.lower())      # lower case
hello world
```

이 외에도 많은 string 함수 제공 (아래 링크 참조)

https://docs.python.org/3.4/library/stdtypes.html#string-methods

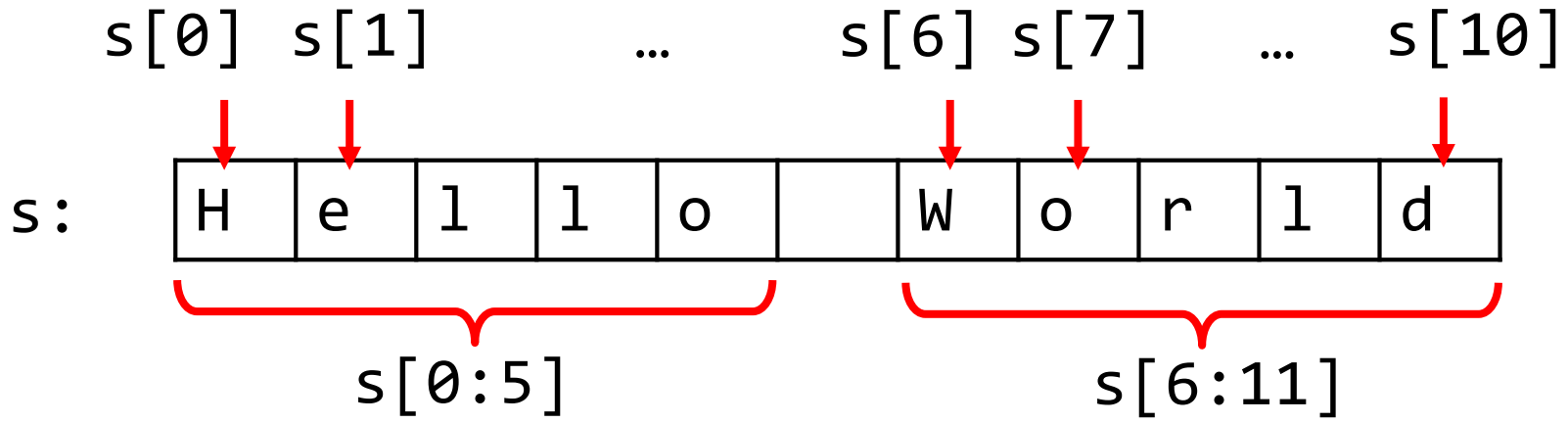
Working with Strings

```
>>> my_name = "John"
>>> full_name = my_name + " Kim" # + for concatenation
>>> print(full_name)
John Kim
>>> print(my_name * 3)           # * for repetition
JohnJohnJohn
>>> my_age = 20
>>> print(my_name + "is " + my_age)
Traceback (most recent call last):
  File "<pyshell#42>", line 1, in <module>
    print(my_name + "is " + my_age)
TypeError: Can't convert 'int' object to str implicitly
>>> print(my_name + "is " + str(my_age))
John is 20
```

Why this error?

Convert int to string

More on Python Strings




```
>>> s = 'Hello World'
>>> s[0]
'H'
>>> s[0:4]
'Hell'
>>> s[0:5]
'Hello'
>>> s[6:11]
'World'
>>> s[6:]
'World'
```

List (1/2)

- 순서대로 나열된 자료들의 집합
- 한 List가 서로 다른 Type의 자료들 포함 가능

```
>>> l1 = [1, 2, 3, 4]
>>> l1[0]
1
>>> l2 = [5, 6, 7, 8]
>>> l = l1 + l2
>>> l
[1, 2, 3, 4, 5, 6, 7, 8]
>>> l = l * 2
>>> l
[1, 2, 3, 4, 5, 6, 7, 8, 1, 2, 3, 4, 5, 6, 7, 8]
```

List (2/2)

```
>>> l = [1, "Hello", 2, "World", 3]
>>> l
[1, 'Hello', 2, 'World', 3]
>>> l.append("Goodbye")
>>> l
[1, 'Hello', 2, 'World', 3, 'Goodbye']
>>> l[3] = 'Korea'
>>> l
[1, 'Hello', 2, 'Korea', 3, 'Goodbye']
```

Tuple

```
>>> t = (1, "Hello", 2, "World", 3)
>>> t
(1, 'Hello', 2, 'World', 3)
>>> t.append("Goodbye")
Traceback (most recent call last):
  File "<pyshell#191>", line 1, in <module>
    t.append("Goodbye")
AttributeError: 'tuple' object has no attribute 'append‘
>>> t[3] = 'Korea'
Traceback (most recent call last):
  File "<pyshell#197>", line 1, in <module>
    t[3] = 'Korea'
TypeError: 'tuple' object does not support item
assignment
```

Cannot append to a tuple

Cannot modify an item inside

Dictionary

- Dictionary: Set of (Key, Values) Pair

```
>>> d = {'John': 177, 'Taylor':173, 'Brenndon':183}
>>> d['John']
177
>>> d['Taylor']
173
>>> d['Brenndon']
183
>>> d = {1:'Apple', 2:'Orange', 3:'Pineapple'}
>>> d[1]
'Apple'
>>> d[2]
'Orange'
>>> d[3]
'Pineapple'
```

Data Type Conversion (1/2)

- 값을 유지한 채 Data Type을 변환
- 다양한 형 변환 함수 제공
  int(), float(), str() 등

```
>>> a = 10
>>> type(a)
<class 'int'>
>>> b = str(a)      # int to str 형변환
>>> b
'10'
>>> type(b)
<class 'str'>'
```

Data Type Conversion (2/2)

```
>>> a = 10
>>> b = '20'
>>> a + b
Traceback (most recent call last):
  File "<pyshell#22>", line 1, in <module>
    a + b
TypeError: unsupported operand type(s) for +: 'int' and
'str'
>>> a + int(b)      # str to int 형변환
30
```

```
>>> a = 10.0
>>> b = "15.5"
>>> a + float(b)    # str to float 형변환
25.5
>>>
```

Questions