Welcome to my class

Presented by…..
Shajeda Yeasmin
Instructor(Computer)
Bangladesh Sweden Polytechnic Institute.
Kaptai. Rangmati.

Name of the Subject & code

Python Programming
Subject Code: 28521

অধ্যায়-৪

পাইথন অপারেটর
(Python Operators)

পাইথন অপারেটর

পাইথনে অপারেটর হলো বিশেষ ধরনের প্রতীক যা গাণিতিক এবং যৌক্তিক (logical) হিসাব-নিকাশে ব্যবহৃত হয়। অপারেটর(+, -, * ইত্যাদি) যে ভ্যালুকে অপারেট করে তাকে অপারেন্ড বলা হয়।

```
x= 3+7
Print(x)
#Output: 10
```

আর ব্যাবহৃত সংখ্যা অর্থাৎ 3 ও 7 হলো অপারেন্ড ।

পাইথনে অপারেটরসমূহঃ

1. এরিথমেটিক অপারেটর(Arithmetic operator)
2. কম্প্যারিজম অপারেটর(Comparison/Relational operator)
3. লজিক্যাল অপারেটর(Logical (Boolean) operator)
4. এসাইনমেন্ট অপারেটর(Assignment operator)
5. বুলিয়ান অপারেটর (Boolean Operator)
6. মেম্বারশিপ অপারেটর (Membership Operator)
7. আইডেন্টিটি অপারেটর (Identity Operator)
8. বিটওয়াইজ অপারেটর(Bitwise operator)

এরিথমেটিক অপারেটর(Arithmetic operator)

গাণিতিক যোগ-বিয়োগ, গুণ-ভাগ ইত্যাদিতে এরিথমেটিক অপারেটর(Arithmetic operator) ব্যবহৃত হয়।

অপারেটর	অর্থ	উদাহরণ
+	দুটি অপারেন্ডের যোগ বা ইউনারি যোগ(unary plus)	x + y
-	বাম অপারেন্ড থেকে ডান অপারেন্ড বিয়োগ বা ইউনারী বিয়োগ	x - y
*	দুটি অপারেন্ডের গুণ	x * y
/	বাম অপারেন্ডকে ডান অপারেন্ড দিয়ে ভাগ(ফলাফল সব সময় দশমিক হবে)	x / y
%	মডিউলাস(Modulus) - বাম অপারেন্ডকে ডান অপারেন্ড দিয়ে ভাগ করে ভাগশেষ	x % y (x/y এর ভাগশেষ)
//	Floor division -ভাগশেষ বাদে পূর্ণ সংখ্যায় ভাগফল	x // y
**	সূচক(Exponent)- ডান অপারেন্ড বাম অপারেন্ড এর সূচক	x**y (x এর সূচক y)

উদাহরণঃ পাইথনে এরিথমেটিক অপারেটর

```
x = 10
y = 4
print('x + y =',x+y)
# Output: x + y = 14
print('x - y =',x-y)
# Output: x - y = 6
print('x * y =',x*y)
# Output: x * y = 40
print('x / y =',x/y)
# Output: x / y = 2.50
print('x // y =',x//y)
# Output: x // y = 3
print('x ** y =',x**y)
# Output: x ** y = 10000
```

কম্প্যারিজম অপারেটর(Comparison/Relational operator)

দুই বা ততোধিক ভ্যালুর মধ্যে তুলনা করতে কম্প্যারিজন অপারেটর ব্যবহৃত হয়। কোনো শর্তের(condition) উপরে ভিত্তিকরে এটি হয় True অথবা False রিটার্ন করে।

পাইথনে কম্প্যারিজন অপারেটরসমূহ

অপারেটর	অর্থ	উদাহরণ
>	Greater Than - বামপক্ষ ডানপক্ষের চেয়ে বড় হলে True হবে।	x > y
<	Less Than - বাম পক্ষ্য ডানপক্ষের চেয়ে ছোট হলে True হবে।	x < y
==	Equal to - বামপক্ষ এবং ডানপক্ষ সমান হলে True হবে।	x == y
!=	Not equal to - বামপক্ষ এবং ডানপক্ষ সমান না হলে True হবে।	x != y
>=	Greater than or equal to - বামপক্ষ ডানপক্ষের চেয়ে বড় বা সমান হলে True হবে।	x >= y
<=	Less than or equal to - বামপক্ষ ডানপক্ষের চেয়ে ছোট বা সমান হলে True হবে।	x <= y

উদাহরণঃ পাইথনে কম্প্যারিজন অপারেটর

```
x = 10
y = 15
print('x > y  is',x>y)
# Output: x > y is False
print('x < y  is',x<y)
# Output: x < y is True
print('x == y is',x==y)
# Output: x == y is False
print('x != y is',x!=y)
# Output: x != y is True
print('x >= y is',x>=y)
# Output: x >= y is False
print('x <= y is',x<=y)
# Output: x <= y is True
```

লজিক্যাল অপারেটর

পাইথনে and, or এবং not অপারেটরসমূহকে লজিক্যাল অপারেটর(Logical operator) বলা হয়। পাইথনে লজিক্যাল অপারেটরসমূহ-

অপারেটর	অর্থ	উদাহরণ
and	উভয় অপারেন্ড true হলে True	a and b
or	যেকোনো একটি অপারেন্ড true হলে True	a or b
not	অপারেন্ড false হলে True	not a

উদাহরণঃ পাইথনে লজিক্যাল অপারেটর

```
a = True
b = False
print('a and b is',a and ab)
# Output: a and b is False

print('a or b is',a or b)
# Output: a or b is True

print('not a is',not a)
# Output: not a is False
```

এসাইনমেন্ট অপারেটর

ভ্যারিয়েবলে ভ্যালু এসাইন বা আরোপ করার জন্য পাইথনে এসাইনমেন্ট অপারেটর(Assignment operator) ব্যব্যহৃত হয়।a = 10 একটি সাধারণ এসাইনমেন্ট অপারেটর যা ডানপাশের ভ্যালু 10 কে বামপাশের ভ্যারিয়েবল a তে এসাইন করে।পাইথনে বিভিন্ন ধরণের অপারেটর আছে যেমন- a += 5 ভ্যারিয়েবলের সাথে ভ্যালু যোগ করে যোগফল পরবর্তীতে একই ভ্যারিয়েবলে এসাইন করে। এটি a = a + 5 এর সমতুল্য।

পাইথনে এসাইনমেন্ট অপারেটর

অপারেটর	উদাহরণ	সমতুল্য
=	x = 5	x = 5
+=	x += 5	x = x + 5
-=	x -= 5	x = x - 5
*=	x *= 5	x = x * 5
/=	x /= 5	x = x / 5
%=	x %= 5	x = x % 5
//=	x //= 5	x = x // 5
**=	x **= 5	x = x ** 5
&=	x &= 5	x = x & 5
|=	x |= 5	x = x | 5
^=	x ^= 5	x = x ^ 5
>>=	x >>= 5	x = x >> 5

বুলিয়ান অপারেটর (Boolean Operator)

পাইথনে সত্য - মিথ্যা (True/False) যাচাই এর ক্ষেত্রে Boolean বা Logical Operator ব্যবহার করা হয়।

পাইথনে Boolean ভ্যালু হল দুইটি -

1. True (T must be Capital Letter)
2. False (F must be Capital Letter)

পাইথনে বুলিয়ান অপারেটর আছে তিনটি - and , or, not ।

ইংরেজিতে and, or, not কে আমরা 'Conjunction' হিসেবে ব্যবহার করি। তাই প্রোগ্রামিং এ এদেরকে 'Logical Conjunction' ও বলা হয়। পাইথনে এরা প্রত্যেকেই কিছু নীতি বা রুলস মেনে চলে।

and এর ক্ষেত্রেঃ

1. True and True is True

ব্যাখ্যাঃ 'and' দ্বারা দুইটি সংখ্যাকে প্রকাশ করা হলে, যদি এরা উভয়ই সত্যি (True) হয় (মানে শূন্য ছাড়া যেকোন সংখ্যা), তবে ফলাফল বা রিটার্নও সত্যি (True) হবে। যেমনঃ

```
>>>x=4 and 4  (4 = True  and 4 =True)

>>> 4                (4 = True)
```

2. True and False is False

ব্যাখ্যাঃ 'and' দ্বারা দুইটি সংখ্যাকে প্রকাশ করা হলে, যদি প্রথম সংখ্যাটি সত্যি(True) হয় আর পরবর্তী সংখ্যাটি মিথ্যা (False) হয় (মানে শূন্য হয়), তবে ফলাফল বা রিটার্নও মিথ্যা (False) হবে। যেমনঃ

```
>>> 4 and 0          (4 = True  and  0=False)
>>> 0                (0 = False)
```

3. False and True is False => 0 and 1 = 0
4. False and False is False => 0 and 0 = 0

Example

```
print(1 and 1)

print(1 and 0)

print(0 and 1)

print(0 and 0)
```

or এর ক্ষেত্রেঃ

1. True or True is True => 1 or 1 = 1
2. True or False is True => 1 or 0 = 1
3. False or True is True => 0 or 1 = 1
4. False or False is False => 0 or 0 = 0

Example

```
print(1 or 1)
print(1 or 0)
print(0 or 1)
print(0 or 0)
```

not এর ক্ষেত্রেঃ

1. not True is False => not 1 = False
2. not False is True => not 0 = True

Example

```
print(not 1)
print(not 0)
```

মেম্বারশিপ অপারেটর (Membership Operator

পাইথনে in এবং not in হচ্ছে মেম্বারশিপ অপারেটর(Membership operator)। কোনো ভ্যালু বা ভ্যারিয়েবল string, list, tuple, set এবং dictionary ক্রমের মধ্যে আছে কি না যাচাই করার জন্য এগুলো ব্যবহৃত হয়।ডিকশনারিতে ভ্যালুর পরিবর্তে কোনো কী(key)-এর উপস্থিতি আছে কিনা যাচাই

করা হয়।পাইথনে মেম্বারশিপ অপারেটর

পারেটর	অর্থ	উদাহরণ
in	ক্রম(sequence)-এর মধ্যে ভ্যলু/ভ্যারিয়েবল থাকলে True হবে।	5 in x
not in	ক্রম(sequence)-এর মধ্যে ভ্যলু/ভ্যারিয়েবল না থাকলে True হবে।	5 not in x

উদাহরণঃ পাইথনে মেম্বারশিপ অপারেটর

```
list = [1, 2, 3, 4, 5 ]
if (2 in list):
    print("2 exist in the list")
else:
    print("2 is not exist in the list")
```

আইডেন্টিটি অপারেটর (Identity Operator)

আইডেন্টিটি অপারেটর দিয়ে দুইটা অবজেক্টের মেমরি লোকেশন একই কিনা, তা যাচাই করা হয়। পাইথনে দুইটা আইডেন্টিটি অপারেটর রয়েছেঃ

1. is
2. is not

আমরা যদি একই ভ্যালু দুইটা ভ্যারিয়েবলের রাখি, মেমরি সেভ করার জন্য ঐ দুইটা ভ্যারিয়েবলে ঐ ভ্যালুটির লোকেশনে পয়েন্ট করে। যেমনঃ

```
a = 20
b = 20
if ( a is b ):
    print ("a and b have same identity")
else:
    print ("a and b do not have same identity")
```

Output : a and b have same identity

দুইটাতে আলাদা ভ্যালু সেট করে রান করে দেখেন কি আউটপুট দিচ্ছে।

```
a = 20
b = 21

if ( a is not b ):
   print ("a and b have different identity")
else:
   print ("a and b do have same identity")
```

উপরের প্রোগ্রাম রান করে দেখলে

আউটপুট দিচ্ছেঃ a and b have different identity

বিটওয়াইজ অপারেটর(Bitwise operator)

পাইথনে সহজেই আমরা বিটওয়াইজ অপারেশন করতে পারি। নিচে বিটওয়াইজ অপারেটর গুলোর লিস্ট দেওয়া হলোঃ

মনেকরি, নিচের টেবিলে x = 10 (বাইনারিতে 0000 1010) এবং y = 4 (বাইনারিতে 0000 0100)

অপারেটর	নাম	বর্ণনা
&	AND	লজিক্যাল AND অপারেটর; অর্থাৎ 1 হবে যদি সকল বিট (Input Line) 1 হয়।
|	OR	লজিক্যাল OR অপারেটর; অর্থাৎ যেকোন একটা বিট (Input Line) 1 হলেই 1 হবে।
^	XOR	লজিক্যাল XOR অপারেটর; অর্থাৎ যদি বিজোড় সংখ্যক বিট (Input Line) 1 হয় তবেই শুধুমাত্র 1 হবে।
~	NOT	লজিক্যাল NOT অপারেটর; অর্থাৎ 0 থাকলে 1 এবং 1 থাকলে 0 হবে।
<<	Binary left shift	বিটগুলোকে বাম দিকে সরিয়ে দেওয়ার জন্য।
>>	Binary right shift	বিটগুলোকে ডান দিকে সরিয়ে দেওয়ার জন্য।

বিটওয়াইজ AND অপারেটরের ব্যবহার-

```
a = 10
b = 4;
print(a&b)
```

বিটওয়াইজ OR(|) অপারেটর এর ব্যবহার-

```
a = 10
b = 4;
print(a|b)
```

বিটওয়াইজ XOR(^) অপারেটর এর ব্যবহার-

```
a = 10
b = 4;
print(a^b)
```

বিটওয়াইজ complement(~) অপারেটর এর ব্যবহার-

যেকোনো সংখ্যা N এর বিটওয়াইজ কমপ্লিমেন্ট হলো -(N+1)

```
print("complement",~10)
print("complement",~-20)
```

বিটওয়াইজ Left Shift(<<) অপারেটর & বিটওয়াইজ Left Shift(>>) অপারেটর

print(a<<2) # Binary Left Shift অপারেটর।

print(a>>2) # Binary Right Shiftঅপারেটর

টীকাঃ এখানে a<<2 এবং a>>2 দিয়ে দুই ঘর বামে এবং ডানে সরানো হয়েছে। a এর ভ্যালু 10 এবং 10 এর বাইনারি 1010; এটাকে দুইঘর বামে সরালে হবে 1010 00 অর্থাৎ 40. আবার এটাকে ডানে সরালে 00 10 অর্থাৎ ২ হবে।

পাইথন অপারেটর প্রেসিডেন্সি ও অ্যাসোসিয়েটিভিটি

অপারেটর প্রেসিডেন্স

সাধারণ গণিতে যেমন যোগ বা বিয়োগের আগে গুন ও ভাগ করে নিতে হয় তেমনি প্রোগ্রামিং -এও এই অপারেটর গুলোর একটা অগ্রাধিকার মূলক নিয়ম আছে। অর্থাৎ সেই নিয়ম মেনেই একটি স্টেটমেন্ট এর মধ্যে থাকা একাধিক অপারেটরের অপারেশন ঘটবে। এটা গণিতের সরল করার নিয়মের সাথেই মিলে যায় অর্থাৎ - প্রথমেই ব্র্যাকেটের কাজ, তারপর পাওয়ার/এক্সপোনেন্ট, অতঃপর গুন ও ভাগ এবং শেষে যোগ ও বিয়োগ।যোগ, বিয়োগ, গুন, ভাগ বাদেও যেহেতু প্রোগ্রামিং -এ আরও কিছু অপারেটর আছে, তাই সেগুলোর অগ্রাধিকারও জেনে রাখা দরকার। যেমন নিচের স্টেটমেন্ট দুটি দেখি,

```
>>> False == False or True
True
>>> False == (False or True)
False
```

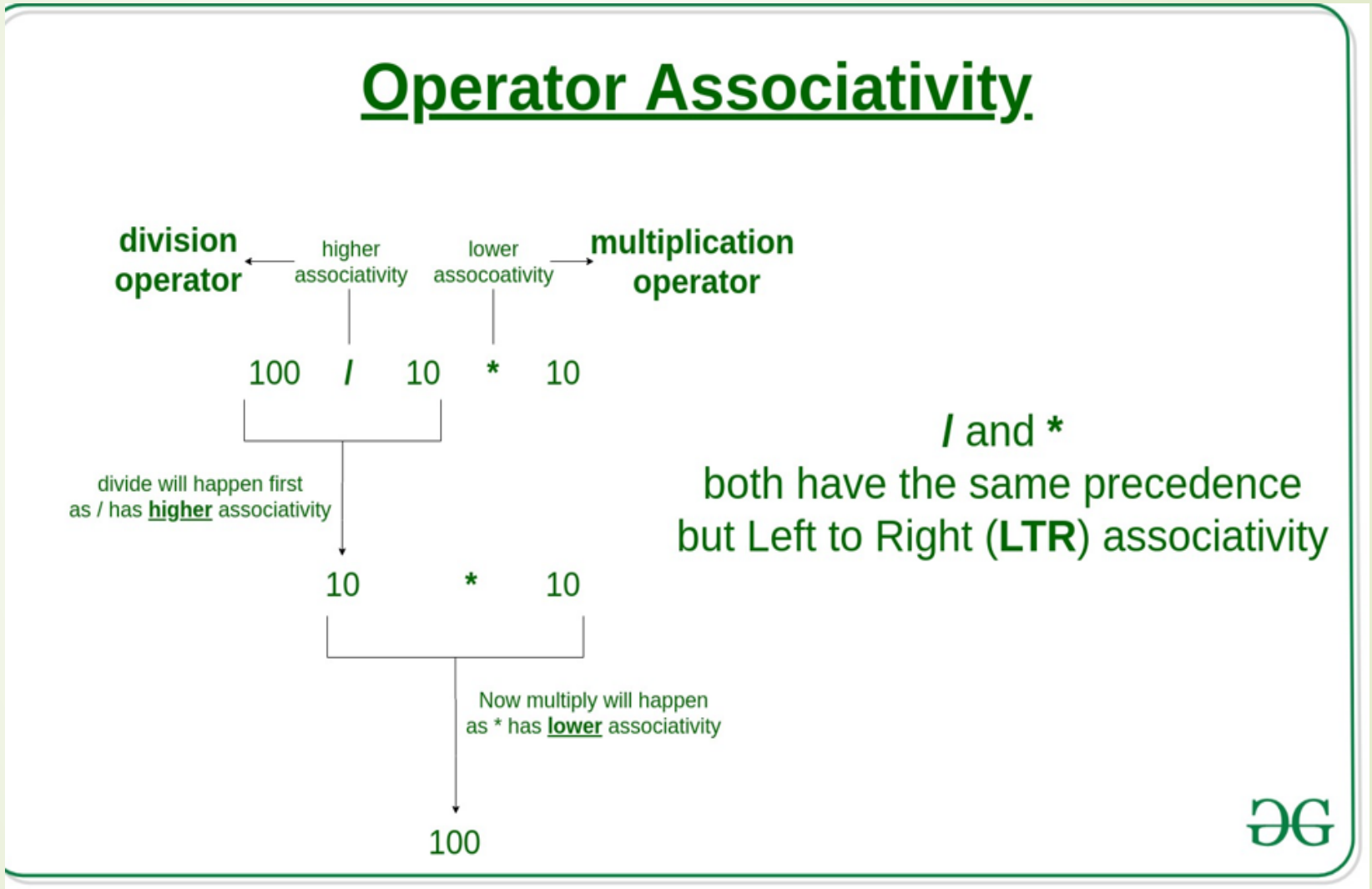
উপরের প্রথম স্টেটমেন্টে == এর অগ্রাধিকার or চেয়ে বেশি। আর নিচের স্টেটমেন্টে or অপারেশন অগ্রাধিকার পেয়েছে কারন এটি একটি বন্ধনীর মধ্যে অবস্থান করছে।

Operator Precedence

The following table lists all of Python's operators, from highest precedence to lowest.

Operator	Description
**	Exponentiation (raise to the power)
~, +, -	Complement, unary plus and minus (method names for the last two are +@ and -@)
*, /, %, //	Multiply, divide, modulo and floor division
+, -	Addition and subtraction
>>, <<	Right and left bitwise shift
&	Bitwise 'AND'
^	Bitwise exclusive 'OR'
|	Bitwise 'OR'
in, not in, is, is not, <, <=, >, >=, !=, ==	Comparison operators, equality operators, membership and identity operators
not	Boolean 'NOT'
and	Boolean 'AND'
or	Boolean 'OR'
=, %=, /=, //=, -=, +=, *=, **=	Assignment operators

পাইথন অপারেটর অ্যাসোসিয়েটিভিটি



ধন্যবাদ