

INF 110 **Discovering Informatics**

More Expressions

Logical Operations

- $x \text{ and } y$
- $x \text{ or } y$
- $\text{not } x$

A	B	A and B	A or B
FALSE	FALSE	FALSE	FALSE
FALSE	TRUE	FALSE	TRUE
TRUE	TRUE	TRUE	TRUE

- $\text{True and True} \rightarrow \text{True}$
- $\text{True or True} \rightarrow \text{True}$

If statements

An if statement uses logical expressions to conditionally evaluate statements

```
if <expr>:  
    statement1  
    statement2  
    ...  
elif <expr>:  
    statement1  
    statement2  
    ...  
else:  
    statement1  
    statement2  
    ...
```

Live Code How's the Weather?

Task: Write an if statement that provides a statement about the weather based on the temperature

Learning Outcomes

- Designing if statements
- Solving problems with comparison operators
- Solving problems with logical operators

Function Composition

```
1 # First we associate the string with a variable
2 x = "Plum"
3
4 # Second we find the length of x
5 x_length = len(x)
6
7 # Third we convert it to binary
8 bin_string = bin(x_length)
9
10 # Fourth, we print the result
11 print(bin_string)
```

Function Composition

By using function composition, we can combine all of those steps into a single line:

```
1 print(bin(len("Hello")))
```

Function Composition

```
1 number = input("Enter a number: ")
2 base = input("Enter its base: ")
3
4 print(int(number, int(base)))
```



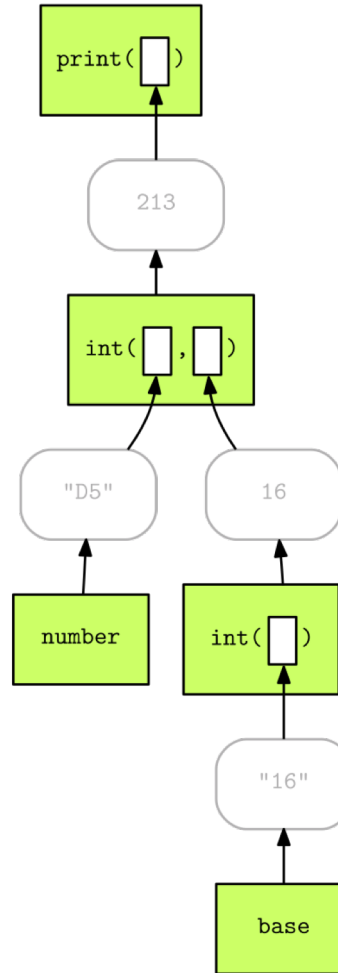
Subexpressions

A logically complete expression that is part of a compound expression.

```
print(bin(len(x) + 3))
```

- What are the subexpressions?
- What wouldn't be a subexpression?

Expression Tree Evaluation



Method Chaining

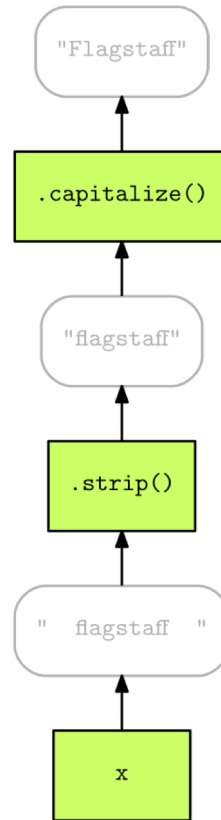
```
1 x = "   flagstaff   "  
2  
3 # Remove the whitespace  
4 x = x.strip()  
5  
6 # Capitalize  
7 x = x.capitalize()  
8  
9 print(x)
```

Method Chaining

The same thing can be accomplished using method chaining:

```
1 x = "   flagstaff   "  
2  
3 print(x.strip().capitalize())
```

Method Chaining Expression Tree



Operator Precedence

Table 0.0.1: *Operator Precedence*

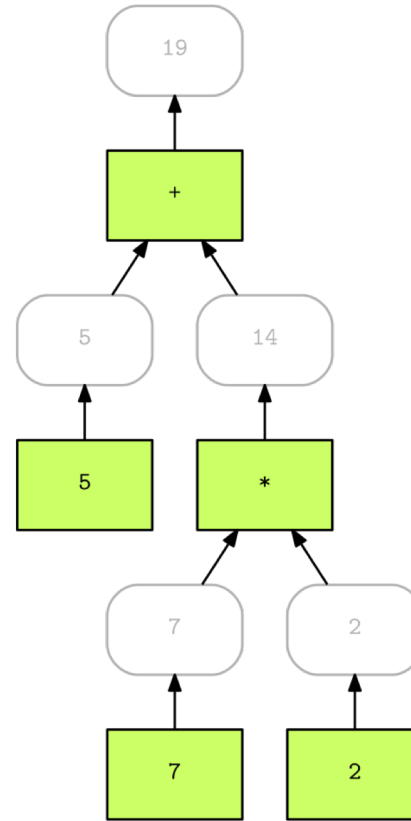
Precedence	Operator Family
High	<code>()</code> , <code>[]</code> , <code>{}</code>
	<code>x[]</code> , <code>x()</code> , <code>x.attribute</code>
	<code>**</code>
	<code>+x</code> , <code>-x</code>
	<code>*</code> , <code>/</code> , <code>//</code> , <code>%</code>
	<code>+</code> , <code>-</code>
	<code>in</code> , <code>not in</code> , <code><</code> , <code><=</code> , <code>></code> , <code>>=</code> , <code>!=</code> , <code>==</code>
	<code>not x</code>
	<code>and</code>
	<code>or</code>
Low	<code>lambda</code>

Operator Precedence

Table 0.0.1: *Operator Precedence Examples*

Example	Fully Qualified Example
$10 - 4 + 2$	$(10 - 4) + 2$
$10 - 4 * 2$	$10 - (4 * 2)$
$p + q * r + s$	$(p + (q * r)) + s$
$p + q * r + s / t$	$(p + (q * r)) + (s / t)$
$p \text{ and } q \text{ or } r \text{ and } s$	$(p \text{ and } q) \text{ or } (r \text{ and } s)$
$p \text{ and } q \text{ or } r \text{ and } s \text{ or } t \text{ and } u$	$((p \text{ and } q) \text{ or } (r \text{ and } s)) \text{ or } (t \text{ and } u)$

Operator Precedence



Augmented Assignment Operators

- $x += y$ means $x = x + y$
- $x -= y$ means $x = x - y$
- $x *= y$ means $x = x * y$
- $x /= y$ means $x = x / y$

Augmented Assignment Operators

```
1 number_of_widgets = 7
2
3 # We can add one the long way..
4 number_of_widgets = number_widgets + 1
5
6 # Or the short way..
7 number_of_widgets += 1
```

Live Code Vinyl Record Sales?

Task: What is the **percent difference** in record sales for these two years?

14.32 million in 2017

13.1 million in 2016

Learning Outcomes

- Solving problems with math operators
- Using data to make inferences

end