

# Data Types and Variables

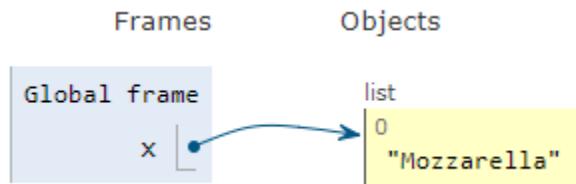
# Python Variables

- no explicit declaration needed
- both value and type may change

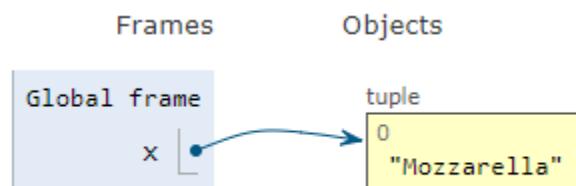
```
>>> x = 5 # int
```



```
>>> x = ["Mozzarella"] # list
```



```
>>> x = ("Mozzarella") # tuple
```



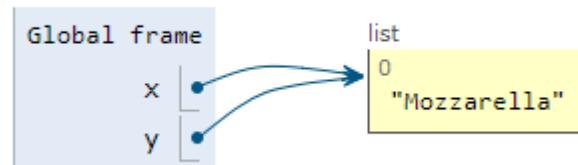
# Python Variables

- not a name for a memory location
  - variables are references to objects

```
>>> x = ["Mozzarella"]
```

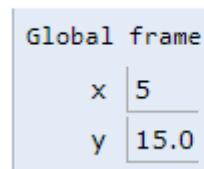


```
>>> y = X
```



```
>>> x = 5
```

```
>>> y = 15.0
```



# Naming Identifiers

- start character:s with alphabetical or ‘\_’
- any alphanumeric or ‘\_’
- no reserved keywords
- case sensitive
- ‘\_’ holds the last computation

```
>>> _123 = 5
>>> !123 = 5
      File "<stdin>", line 1
          !123 = 5
              ^
SyntaxError: invalid syntax
```

```
>>> _
5
```

# Python Reserved Keywords

- should name variables by keywords

```
>>> help()  
help> keywords
```

and	elif	if	print
as	else	import	raise
assert	except	in	return
break	exec	is	try
class	finally	lambda	while
continue	for	not	with
def	from	or	yield
del	global	pass	

```
help> quit
```

```
>>>
```

# *None* Object

- reserved keyword
- special object – used for null objects
- not a zero or a null pointer!!!
- can be used as a (default) argument
- comparison: >>> if x is *None*:

```
>>> x = None  
>>> y = ""  
>>> z = (x is None)  
>>> w = (y is None)
```

## Frames

Global frame	
x	None
y	""
z	True
w	False

# Standard Data Types

(1) number

```
>>> x = 5
```

(2) string

```
>>> x = "Mozzarella"
```

(3) list

```
>>> x = ["Mozzarella"]
```

(4) dictionary

```
>>> x = {1:"Mozzarella", 2: "Brie"}
```

(5) set

```
>>> x = {"Mozzarella"}
```

(6) tuple

```
>>> x = ("Mozzarella", )
```

# Python Integers

- unlimited length

```
>>> x = 999999999999999999999999  
>>> x  
>>> 999999999999999999999999
```

- four types (base = 10, 16, 8, 2)

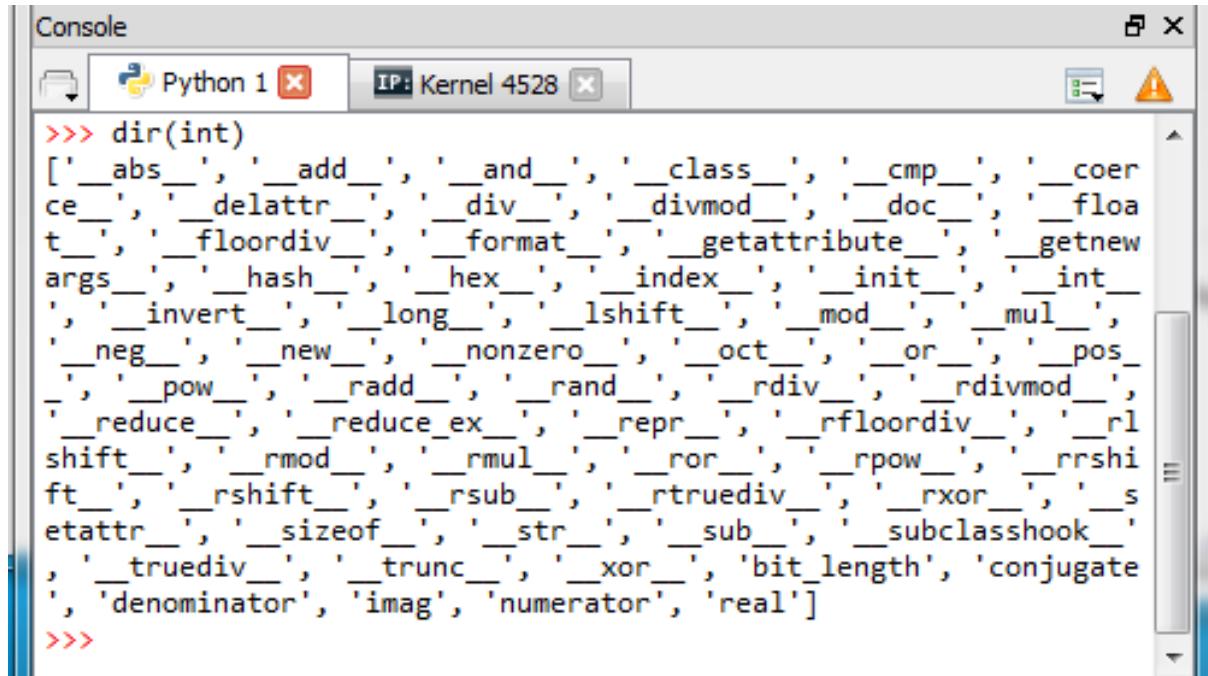
```
>>> x = 100                      # normal integer  
>>> y = 0x40EF                 # hex literal  
>>> z = 0o123                   # octal literal  
>>> w = 0b101                   # binary literal
```

- conversion functions

```
>>> print( hex(x), oct(x), bin(x) )  
>>> ('0x64', '0144', '0b1100100')
```

# *int* Type Methods

>>> **dir(int)**



```
Console
Python 1 IP: Kernel 4528
>>> dir(int)
['__abs__', '__add__', '__and__', '__class__', '__cmp__', '__coerce__',
 '__delattr__', '__div__', '__divmod__', '__doc__', '__float__',
 '__floordiv__', '__format__', '__getattribute__', '__getnewargs__',
 '__hash__', '__hex__', '__index__', '__init__', '__int__',
 '__invert__', '__long__', '__lshift__', '__mod__', '__mul__',
 '__neg__', '__new__', '__nonzero__', '__oct__', '__or__', '__pos__',
 '__pow__', '__radd__', '__rand__', '__rdiv__', '__rdivmod__',
 '__reduce__', '__reduce_ex__', '__repr__', '__rfloordiv__',
 '__rlshift__', '__rmod__', '__rmul__', '__ror__', '__rpow__',
 '__rrshift__', '__rsub__', '__rtruediv__', '__rxor__', '__setattr__',
 '__sizeof__', '__str__', '__sub__', '__subclasshook__',
 '__truediv__', '__trunc__', '__xor__', 'bit_length', 'conjugate',
 'denominator', 'imag', 'numerator', 'real']
```

>>> **int.\_\_abs\_\_(-6)**

>>> **6**

>>> **int.\_\_hex\_\_(100)**

>>> **'0x64'**

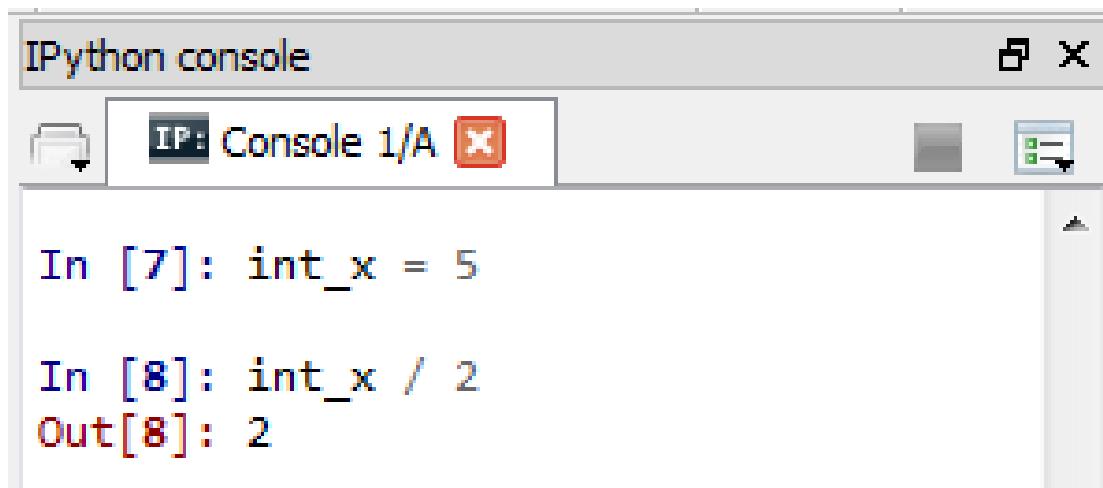
# Bitwise Operations

- defined for integer types only
- assume n=2
- assume (in binary) x = 1010, y = 1100

operation	result	example
$x \mid y$	(bitwise) x or y	1110
$x \wedge y$	(bitwise) excl. or	0110
$x \& y$	(bitwise) x and y	1000
$x \ll n$	shift x left by n	1000
$x \gg n$	shift right by n bits	0010
$\sim x$	invert bits	0101

# Examples of Numbers

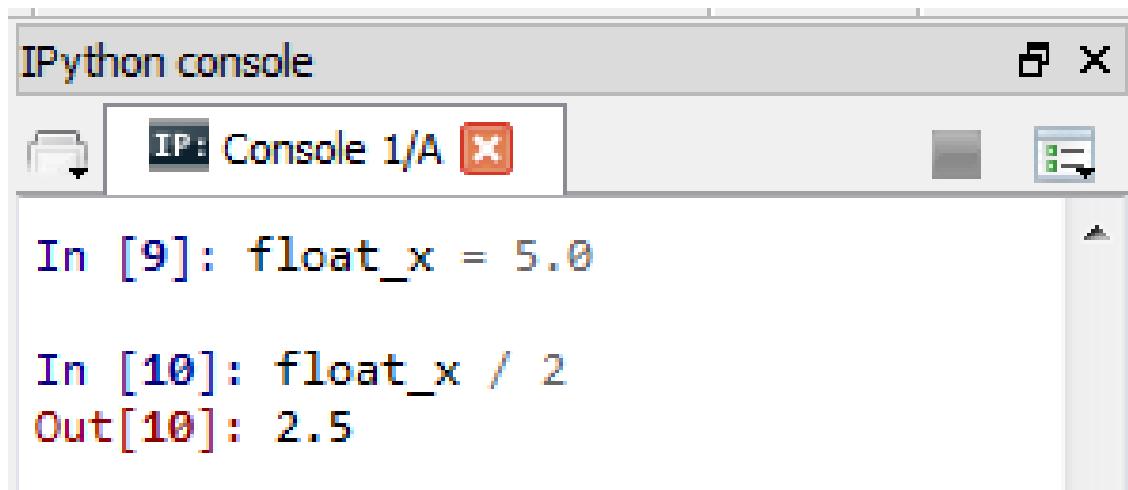
- integers



The screenshot shows an IPython console window titled "IPython console". The tab bar indicates "IP: Console 1/A". The console area displays the following code and output:

```
In [7]: int_x = 5
In [8]: int_x / 2
Out[8]: 2
```

- floating point



The screenshot shows an IPython console window titled "IPython console". The tab bar indicates "IP: Console 1/A". The console area displays the following code and output:

```
In [9]: float_x = 5.0
In [10]: float_x / 2
Out[10]: 2.5
```

- why do we get different results?

# Floating and Complex Numbers

- floating point numbers (many formats)

```
>>> x = 15.491  
>>> y = 2.718e-5
```

- complex numbers (real + imaginary \* j)

```
>>> x = 2 + 2j  
>>> x * x  
>>> 8j  
>>> y = 0 + 1j  
>>> y * y  
>>> -1 + 0j
```

# Type Conversions

- floating point numbers (many formats)

```
>>> x = 15.491  
>>> y = 2.718e-5
```

- complex numbers (real + imaginary \* j)

```
>>> x = 2 + 2j  
>>> x * x  
>>> 8j  
>>> y = 0 + 1j  
>>> y * y  
>>> -1 + 0j
```

# *float* Type Methods

>>> **dir(float)**

```
Console
Python 1 IP: Kernel 4528
>>> dir(float)
['__abs__', '__add__', '__class__', '__coerce__', '__delattr__',
 '__div__', '__divmod__', '__doc__', '__eq__', '__float__', '__floordiv__',
 '__format__', '__ge__', '__getattribute__', '__getformat__',
 '__getnewargs__', '__gt__', '__hash__', '__init__', '__int__',
 '__le__', '__long__', '__lt__', '__mod__', '__mul__', '__ne__',
 '__neg__', '__new__', '__nonzero__', '__pos__', '__pow__',
 '__radd__', '__rdiv__', '__rdivmod__', '__reduce__', '__reduce_ex__',
 '__repr__', '__rfloordiv__', '__rmod__', '__rmul__', '__rpow__',
 '__rsub__', '__rtruediv__', '__setattr__', '__setformat__',
 '__sizeof__', '__str__', '__sub__', '__subclasshook__', '__truediv__',
 '__trunc__', 'as_integer_ratio', 'conjugate', 'fromhex', 'hex',
 '__imag__', 'is_integer', 'real']
>>>
```

Console History log  
Permissions RW End-of-CRLF Encoding UTF-8 Line 8 Col 1 Memory 56 %

>>> **float.is\_integer(5.0)**

True

>>> **float.is\_integer(5.5)**

False

# Numeric Operators

- common operators

```
>>> 10 + 3
```

```
13
```

```
>>> 10 - 3
```

```
7
```

```
>>> 10 * 3
```

```
30
```

```
>>> 10 ** 3          # exponentiation
```

```
1000
```

```
>>> 10 / 3
```

```
3.33333333333333
```

```
>>> 2 + 2 * 2
```

# Division in Python

- “true” division

```
>>> # not needed in Python 3.x
>>> from __future__ import division
>>> 10 / 3
3.333333333333335
>>> -10 / 3
-3.333333333333335
```

- “floor” division

```
>>> 10 // 3
3
>>> -10 // 3
-4
```

# Modular Arithmetic

- numbers “wrap around” upon reaching certain value (“modulus”)
- in math:  $a = b \pmod{n}$
- remainder when divided by modulus

```
>>> 10 % 3
```

```
1
```

```
>>> 11 % 3
```

```
2
```

```
>>> 12 % 3
```

```
0
```

```
>>> 13 % 3
```

```
1
```

```
>>> -10 % 3
```

```
2
```

```
>>> -2 % 3
```

```
1
```

# Types of Operator

(1) arithmetic

```
>>> x = 5 + 7
```

(2) comparison

```
>>> x < y
```

(3) assignment

```
>>> x += 5
```

(4) logical

```
>>> (x and y)
```

(5) bitwise

```
>>> x | y
```

(6) membership

```
>>> x in y
```

(7) identity ( $\text{id}(x)=\text{id}(y)$ )

```
>>> x is y
```

# Augmented Assignments

- simplified form for common operations
- note: no spaces before '='

```
>>> x = 10
>>> y = 5
>>> x += y      # x = x + y
>>> x -= y      # x = x - y
>>> x *= y      # x = x * y
>>> x /= y      # x = x / y    (floating)
>>> x //= y     # x = x // y   (integer)
>>> x **= y     # x = x ** y
```

# Operator Precedence

- (1) exponentiation
- (2) multiplication and division
- (3) addition and subtraction

- same precedence: from left to right
- use brackets to enforce precedence

```
>>> (1 + 2) * 3  
9  
>>> 1 + 2 * 3  
7  
>>> 1 + 2 * 3 + 2 ** 3  
15  
>>> 3 - 1 - 2  
0  
>>> 3 - (1 - 2)  
4
```

# Review Problems

# Interview Problem

- What is the output of the following code:

```
>>> a = 1
```

```
>>> a, b = a + 1, a + 1
```

```
>>> print a
```

```
>>> print b
```

# Interview Problem

- explain the ‘//’ operator

# Interview Problem

- how to convert a number to a string?

# Interview Problem

- how does the ternary operator work?

# Interview Problem

- given a list of prices, compute a list of last digits (in cents position)

# Interview Problem

- write a program to swap two numbers

# Programming Exercise

The screenshot shows a user interface for a programming exercise. At the top, there are navigation buttons: 'PREV' and 'NEXT' in green rounded rectangles, followed by a 'Workbench' button in blue, and a help icon with a question mark and a gear symbol. Below these are two tabs: 'WORK AREA' and 'SOLUTIONS'. A 'Content Support' button is located in the top right corner of the main area. The main content area contains the following text:

Assign 7 to a variable named seven.

# **Programming Exercise**

## **Worksheet**

**this page is intentionally left blank**

# Programming Exercise

The screenshot shows a user interface for a programming exercise. At the top, there are navigation buttons: 'PREV' and 'NEXT' on the left, and 'Workbench' in the center. To the right of 'Workbench' are two icons: a question mark and a gear. Below these are two tabs: 'WORK AREA' and 'SOLUTIONS'. A green button labeled 'Content Support' is located below the tabs. The main area contains the following text:

Define a variable **temperature** and  
make it refer to 98.6.

# **Programming Exercise**

## **Worksheet**

**this page is intentionally left blank**

# Programming Exercise

The screenshot shows a user interface for a programming exercise. At the top, there are navigation buttons: 'PREV' and 'NEXT' on the left, 'Workbench' in the center, and a help icon ('?') and settings icon ('⚙️') on the right. Below these are two tabs: 'WORK AREA' and 'SOLUTIONS'. A 'Content Support' button is located above the main workspace. The main workspace contains the following text:

Define a variable precise and make  
it refer to 1.09388641.

# **Programming Exercise**

## **Worksheet**

**this page is intentionally left blank**

# Programming Exercise

The screenshot shows a user interface for a programming exercise. At the top, there are navigation buttons: '1 PREV' and 'NEXT 2'. To the right of these are links for 'Workbench', '?', and a gear icon. Below the navigation is the title 'Exercise 51019 –'. Underneath the title are two tabs: 'WORK AREA' and 'SOLUTIONS'. A green button labeled 'Content Support' is located above the main workspace. The workspace contains the following text: 'Assign 8 to a variable named eight.'

# **Programming Exercise**

## **Worksheet**

**this page is intentionally left blank**

# Programming Exercise

The screenshot shows a user interface for a programming exercise. At the top, there are navigation buttons: 'PREV' and 'NEXT' on the left, 'Workbench' in the center, and a help icon with a gear on the right. Below these are two tabs: 'WORK AREA' and 'SOLUTIONS'. A 'Content Support' button is located below the tabs. The main content area contains the following text:

Assign the average of the values in the variables a, b, and c to a variable avg. Assume that the variables a, b, and c have already been assigned a value, but do not assume that the values are all floating-point. The average should be a floating-point value.

# **Programming Exercise**

## **Worksheet**

**this page is intentionally left blank**

# Programming Exercise

The screenshot shows a user interface for a programming exercise. At the top, there are navigation buttons: 'PREV' and 'NEXT' on the left, 'Workbench' in the center, and a help icon ('?') and settings icon ('⚙️') on the right. Below these are two tabs: 'WORK AREA' and 'SOLUTIONS'. A green button labeled 'Content Support' is located above the main content area. The main content area contains the following text:

Calculate the BMI of a person using the formula  $BMI = (\text{Weight in Pounds} / ((\text{Height in inches}) \times (\text{Height in inches})) \times 703$  and assign the value to the variable `bmi`. Assume the value of the `weight in pounds` has already been assigned to the variable `w` and the value of the `height in inches` has been assigned to the variable `h`. Take care to use floating-point division.

# **Programming Exercise**

## **Worksheet**

**this page is intentionally left blank**

# Programming Exercise

The screenshot shows a user interface for a programming exercise. At the top, there are navigation buttons: 'PREV' and 'NEXT' on the left, 'Workbench' in the center, and a question mark icon and gear icon on the right. Below these is the title 'Exercise 51029 —'. Underneath the title are two tabs: 'WORK AREA' and 'SOLUTIONS'. A green button labeled 'Content Support' is located above the main content area. The main content area contains the following text:

Write an **expression** that  
computes the **sum** of two **variables**  
`verbal_score` and `math_score`  
(assume that both have already  
been defined).

# **Programming Exercise**

## **Worksheet**

**this page is intentionally left blank**

# Programming Exercise

The screenshot shows a user interface for a programming exercise. At the top, there are navigation buttons: 'PREV' and 'NEXT' on the left, 'Workbench' in the center, and a help icon ('?') and settings icon ('⚙️') on the right. Below these is the title 'Exercise 51030 —'. Underneath the title are two tabs: 'WORK AREA' (which is selected) and 'SOLUTIONS'. A green button labeled 'Content Support' is located below the tabs. The main area contains the following text:

Given the variables  
taxable\_purchases and  
tax\_free\_purchases (which already  
have been defined), write an  
**expression** corresponding to the  
**total amount** purchased.

# **Programming Exercise**

## **Worksheet**

**this page is intentionally left blank**

# Programming Exercise

The screenshot shows a user interface for a programming exercise. At the top, there are navigation buttons: 'PREV' and 'NEXT' in green rounded rectangles, followed by a 'Workbench' button, a help icon (question mark), and a settings gear icon. Below these are two tabs: 'WORK AREA' and 'SOLUTIONS'. A green button labeled 'Content Support' is located on the right side of the main area. The main content area contains the following text:

Exercise 51031 –

Write an **expression** that computes the **difference** of the variables `ending_time` and `starting_time`.

# **Programming Exercise**

## **Worksheet**

**this page is intentionally left blank**

# Programming Exercise

The screenshot shows a user interface for a programming exercise. At the top, there are navigation buttons: 'PREV' and 'NEXT' on the left, and 'Workbench' in the center. To the right of 'Workbench' are two icons: a question mark and a gear. Below these buttons, the title 'Exercise 51032 —' is displayed. Underneath the title, there are two tabs: 'WORK AREA' and 'SOLUTIONS'. The 'WORK AREA' tab is currently selected. In the main content area, there is a green button labeled 'Content Support'. The main text area contains the following instruction:

Given the variables  
full\_admission\_price and  
discount\_amount (already defined),  
write an **expression** corresponding  
to the price of a **discount**  
**admission**.

# **Programming Exercise**

## **Worksheet**

**this page is intentionally left blank**

# Programming Exercise

The screenshot shows a user interface for a programming exercise. At the top, there are navigation buttons: 'PREV' and 'NEXT' on the left, 'Workbench' in the center, and a question mark icon and a gear icon on the right. Below these are two tabs: 'WORK AREA' (highlighted in green) and 'SOLUTIONS'. A 'Content Support' button is located below the tabs. The main area contains the following text:

Given the variable `price_per_case`,  
write an **expression** corresponding  
to the price of a **dozen cases**.

# **Programming Exercise**

## **Worksheet**

**this page is intentionally left blank**

# Programming Exercise

The screenshot shows a user interface for a programming exercise. At the top, there are navigation buttons: 'PREV' and 'NEXT' on the left, 'Workbench' in the center, and a help icon ('?') and settings icon ('gear') on the right. Below these are two tabs: 'WORK AREA' (selected) and 'SOLUTIONS'. A 'Content Support' button is located below the tabs. The main area contains the following text:

Given the variables  
cost\_of\_bus\_rental and  
max\_bus\_riders, write an  
**expression** corresponding to the  
**cost per rider** (assuming the bus  
is full).

# **Programming Exercise**

## **Worksheet**

**this page is intentionally left blank**

# Programming Exercise

The screenshot shows a user interface for a programming exercise. At the top, there are navigation buttons: 'PREV' and 'NEXT' on the left, and 'Workbench' in the center. To the right of 'Workbench' are a question mark icon and a gear icon. Below these are two tabs: 'WORK AREA' and 'SOLUTIONS'. A green button labeled 'Content Support' is located near the bottom of the interface. The main area contains the following text:

Write an **expression** that  
computes the **remainder** of the  
variable principal when divided by  
the variable divisor. (Assume that  
each is associated with an int.)

# **Programming Exercise**

## **Worksheet**

**this page is intentionally left blank**

# Programming Exercise

The screenshot shows a user interface for a programming exercise. At the top, there are navigation buttons: 'PREV' and 'NEXT' in green rounded rectangles, followed by a 'Workbench' button in blue and orange, and a help/gear icon. Below these are two tabs: 'WORK AREA' and 'SOLUTIONS'. A 'Content Support' button is located above a scrollable text area. The text area contains the following instruction:

Write an **expression** that computes the **average** of the values 12 and 40, and assign it to the **variable** avg, which has already been defined.

# **Programming Exercise**

## **Worksheet**

**this page is intentionally left blank**

# Programming Exercise

The screenshot shows a user interface for a programming exercise. At the top, there are navigation buttons: 'PREV' and 'NEXT' on the left, 'Workbench' in the center, and a help icon ('?') and settings icon ('⚙️') on the right. Below these is the title 'Exercise 51037 —'. Underneath the title are two tabs: 'WORK AREA' and 'SOLUTIONS', with 'SOLUTIONS' being the active tab. A green button labeled 'Content Support' is located above the main content area. The main content area contains the following text:

Write an **expression** that computes the **average** of the variables `exam1` and `exam2` (both already assigned values).

# **Programming Exercise**

## **Worksheet**

**this page is intentionally left blank**

# Programming Exercise

The screenshot shows a user interface for a programming exercise. At the top, there are navigation buttons: 'PREV' and 'NEXT'. To the right of these are links for 'Workbench', '?', and a gear icon. Below the navigation is the title 'Exercise 51043 —'. Underneath the title are two tabs: 'WORK AREA' and 'SOLUTIONS'. A green button labeled 'Content Support' is located above the main content area. The main content area contains the following text:

Define two variables, one named length making it refer to 3.5 and the other named width making it refer to 1.55.

# **Programming Exercise**

## **Worksheet**

**this page is intentionally left blank**

# Programming Exercise

The screenshot shows a user interface for a programming exercise. At the top, there are navigation buttons: 'PREV' and 'NEXT' on the left, 'Workbench' in the center, and a help icon ('?') and settings icon ('⚙️') on the right. Below these is the title 'Exercise 51044 —'. Underneath the title are two tabs: 'WORK AREA' and 'SOLUTIONS', with 'WORK AREA' being the active tab. A green button labeled 'Content Support' is located above the main content area. The main content area contains the following text:

Write an **expression** that  
computes the **sum** of two variables  
total1 and total2, which have  
already been defined.

# **Programming Exercise**

## **Worksheet**

**this page is intentionally left blank**

# Programming Exercise

The screenshot shows a user interface for a programming exercise. At the top, there are navigation buttons: 'PREV' and 'NEXT' on the left, a 'Workbench' tab in the center, and a help icon ('?') and settings icon ('⚙️') on the right. Below the navigation is the title 'Exercise 51045 —'. Underneath the title are two tabs: 'WORK AREA' and 'SOLUTIONS'. A green button labeled 'Content Support' is located above the main content area. The main content area contains the following text:

Write an **expression** that  
computes the **difference** of two  
variables `sales_summer` and  
`sales_spring`, both of which have  
already defined.

# **Programming Exercise**

## **Worksheet**

**this page is intentionally left blank**

# Programming Exercise

The screenshot shows a user interface for a programming exercise. At the top, there are navigation buttons: 'PREV' and 'NEXT' on the left, 'Workbench' in the center, and a help icon with a question mark and a gear icon on the right. Below these are two tabs: 'WORK AREA' and 'SOLUTIONS'. A 'Content Support' button is located above a large text area. The main text area contains the following instruction:

You are given two variables, already defined, named `total_weight`, containing the weight of a `shipment`, and `weight_of_box`, containing the weight of the `box` in which a `product` is shipped. Write an **expression** that calculates the **net weight** of the `product`.

# **Programming Exercise**

## **Worksheet**

**this page is intentionally left blank**

# Programming Exercise

The screenshot shows a user interface for a programming exercise. At the top, there are navigation buttons: 'PREV' and 'NEXT' on the left, 'Workbench' in the center, and a question mark icon and a gear icon on the right. Below these are two tabs: 'WORK AREA' and 'SOLUTIONS', with 'SOLUTIONS' currently selected. A green button labeled 'Content Support' is located above the main text area. The main text area contains the following instruction:

You are given two variables, both already defined. One is named price and is associated with a float and is the price of an order. The other is total\_number and is associated with an int and is the number of orders. Write an **expression** that calculates the **total price for all orders**.

# **Programming Exercise**

## **Worksheet**

**this page is intentionally left blank**

# Programming Exercise

The screenshot shows a user interface for a programming exercise. At the top, there are navigation buttons: 'PREV' and 'NEXT' in green rounded rectangles, followed by a 'Workbench' button with a gear icon, and a help icon (question mark) and settings icon. Below these are two tabs: 'WORK AREA' and 'SOLUTIONS'. A 'Content Support' button is located above a large text area. The main text area contains the following question:

Given two variables `matric_age` and `grad_age`, write a statement that makes the associated value of `grad_age` 4 more than that of `matric_age`.

# **Programming Exercise**

## **Worksheet**

**this page is intentionally left blank**

# Programming Exercise

The screenshot shows a user interface for a programming exercise. At the top, there are navigation buttons: 'PREV' and 'NEXT' on the left, 'Workbench' in the center, and a question mark icon and gear icon on the right. Below these are two tabs: 'WORK AREA' and 'SOLUTIONS'. A 'Content Support' button is located above a large text area. The main text area contains the following instructions:

Variables i and j each have associated values. Swap them, so that i becomes associated with j's original value, and j becomes associated with i's original value. You can use two more variables itemp and jtemp.

# **Programming Exercise**

## **Worksheet**

**this page is intentionally left blank**

# Programming Exercise

The screenshot shows a user interface for a programming exercise. At the top, there are navigation buttons: 'PREV' and 'NEXT' in green rounded rectangles, followed by a 'Workbench' button in blue and red, and a help icon with a question mark and gear in green. Below these are two tabs: 'WORK AREA' and 'SOLUTIONS'. A green button labeled 'Content Support' is located above a large text area. The main text area contains the following question:

Given two already defined variables, i and j, write a statement that swaps their associated values.

# **Programming Exercise**

## **Worksheet**

**this page is intentionally left blank**

# Programming Exercise

The screenshot shows a user interface for a programming exercise. At the top, there are navigation buttons: 'PREV' and 'NEXT' in green rounded rectangles, followed by a 'Workbench' button in blue, and a help icon with a question mark and a gear icon. Below these are two tabs: 'WORK AREA' and 'SOLUTIONS'. A 'Content Support' button is located above the main content area. The main content area contains the following text:

Given two variables,  
first\_place\_winner and  
second\_place\_winner, write some  
code that swaps their associated  
values. Use any additional variables  
as necessary.

# **Programming Exercise**

## **Worksheet**

**this page is intentionally left blank**

# Programming Exercise

The screenshot shows a user interface for a programming exercise. At the top, there are navigation buttons: 'PREV' (with a left arrow icon) and 'NEXT' (with a right arrow icon). To the right of these are the words 'Workbench' and icons for help ('?') and settings ('⚙️'). Below the navigation bar, the title 'Exercise 51171 —' is displayed. Underneath the title are two tabs: 'WORK AREA' and 'SOLUTIONS'. A green button labeled 'Content Support' is located below the tabs. The main content area contains the following text:

Given two variables, already defined, `best_value` and `second_best_value`, write a statement that **swaps** their associated values.

# **Programming Exercise**

## **Worksheet**

**this page is intentionally left blank**

# Programming Exercise

The screenshot shows a user interface for a programming exercise. At the top, there are navigation buttons: 'PREV' and 'NEXT' on the left, and 'Workbench' in the center. To the right of 'Workbench' are two icons: a question mark and a gear. Below these buttons, the title 'Exercise 51249 —' is displayed. Underneath the title are two tabs: 'WORK AREA' and 'SOLUTIONS'. The 'WORK AREA' tab is currently selected. In the main content area, there is a green button labeled 'Content Support'. Below this button, a text box contains the following problem statement:

Assume there is a variable, h already assigned a positive integer value. Write the code necessary to assign its square to the variable g. For example, if h had the value 8 then g would get the value 64.

# **Programming Exercise**

## **Worksheet**

**this page is intentionally left blank**

# Programming Exercise

The screenshot shows a user interface for a programming exercise. At the top, there are navigation buttons: 'PREV' and 'NEXT' in green rounded rectangles, followed by a 'Workbench' button in blue, and a help/gear icon. Below these are two tabs: 'WORK AREA' and 'SOLUTIONS'. A 'Content Support' button is located above a scrollable text area. The main text area contains the following question:

The dimensions (width and length) of room1 have been read into two variables: width1 and length1. The dimensions of room2 have been read into two other variables: width2 and length2. Write a single expression whose value is the total area of the two rooms.

# **Programming Exercise**

## **Worksheet**

**this page is intentionally left blank**

# Programming Exercise

The screenshot shows a user interface for a programming exercise. At the top, there are navigation buttons: 'PREV' and 'NEXT' on the left, 'Workbench' in the center, and a help icon with a gear symbol on the right. Below these are two tabs: 'WORK AREA' (highlighted in blue) and 'SOLUTIONS'. A green button labeled 'Content Support' is located above the main text area. The main text area contains the following question:

A wall has been built with two pieces of sheetrock, a smaller one and a larger one. The length of the smaller one is stored in the variable `small`. Similarly, the length of the larger one is stored in the variable `large`. Write a single expression whose value is the length of this wall.

# **Programming Exercise**

## **Worksheet**

**this page is intentionally left blank**

# Programming Exercise

The screenshot shows a user interface for a programming exercise. At the top, there are navigation buttons: 'PREV' and 'NEXT' in green rounded rectangles, a 'Workbench' button in blue, and a help icon with a question mark and gear in green. Below these are two tabs: 'WORK AREA' and 'SOLUTIONS'. A green button labeled 'Content Support' is also visible. The main area contains the following text:

Each of the walls of a room with square dimensions has been built with two pieces of sheetrock, a smaller one and a larger one. The length of all the smaller ones is the same and is stored in the variable small. Similarly, the length of all the larger ones is the same and is stored in the variable large. Write a single expression whose value is the total area of this room. DO NOT use any method invocations.

# **Programming Exercise**

## **Worksheet**

**this page is intentionally left blank**

# Programming Exercise

The screenshot shows a user interface for a programming exercise. At the top, there are navigation buttons: 'PREV' and 'NEXT' in green rounded rectangles, followed by 'Workbench' in blue text, and a help icon ('?') and settings icon ('gear') in green. Below these are two tabs: 'WORK AREA' and 'SOLUTIONS'. A 'Content Support' button is located above a large text area. The main text area contains the following problem statement:

In mathematics, the Nth harmonic number is defined to be  $1 + 1/2 + 1/3 + 1/4 + \dots + 1/N$ . So, the first harmonic number is 1, the second is 1.5, the third is 1.8333... and so on. Write an expression whose value is the 8th harmonic number.

# **Programming Exercise**

## **Worksheet**

**this page is intentionally left blank**

# Programming Exercise

The screenshot shows a software interface for a programming exercise. At the top, there are navigation buttons: 'PREV' and 'NEXT' on the left, 'Workbench' in the center, and a question mark icon and a gear icon on the right. Below the navigation bar, the title 'Exercise 51876' is displayed. Underneath the title, there are two tabs: 'WORK AREA' and 'SOLUTIONS'. A green button labeled 'Content Support' is located above the main content area. The main content area contains the following text:

In mathematics, the Nth harmonic number is defined to be  $1 + 1/2 + 1/3 + 1/4 + \dots + 1/N$ . So, the first harmonic number is 1, the second is 1.5, the third is 1.83333... and so on. Assume that n is an integer variable whose value is some positive integer N. Assume also that hn is a variable whose value is the Nth harmonic number. Write an expression whose value is the  $(N+1)$ th harmonic number.

# **Programming Exercise**

## **Worksheet**

**this page is intentionally left blank**

# Programming Exercise

The screenshot shows a user interface for a programming exercise. At the top, there are navigation buttons: 'PREV' and 'NEXT' on the left, 'Workbench' in the center, and a question mark icon and a gear icon on the right. Below these are two tabs: 'WORK AREA' and 'SOLUTIONS'. A green button labeled 'Content Support' is located near the bottom of the tabs. The main content area contains the following text:

Assume that `price` is an integer variable whose value is the price (in US currency) in cents of an item. Assuming the item is paid for with a minimum amount of change and just single dollars, write an expression for the amount of change (in cents) that would have to be paid.

# **Programming Exercise**

## **Worksheet**

**this page is intentionally left blank**