HÁSKÓLINN Í REYKJAVÍK
REYKJAVIK UNIVERSITY

Replicode tutorial: Hello World example

Helgi Pall Helgason
Replicode source code by Eric Nivel
CADIA / Reykjavik University

In this tutorial

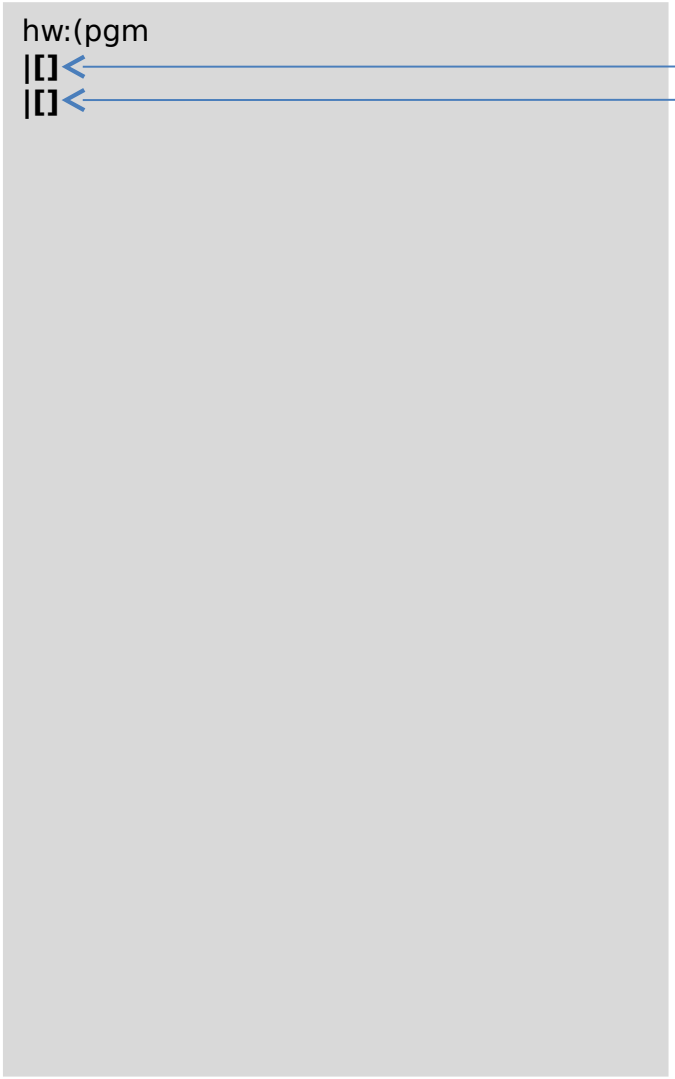
- The Hello World example (included with the Replicode source code) is explained step by step
- Concepts, commands and syntax are explained as they occur
- Explanation of how to interpret the displayed output (which is the decompiled system image) is given

Example #1: Hello world (test.10.replicode)

Definition of a new program starts. Similar to a class definition in traditional OO programming languages.

This program is given a label („hw“) so it may be referred to later.

*In Replicode, **[]** is a set and **|[]** is an empty set*



There are no template arguments for this program. This is indicated by the empty set.

This program requires no inputs. Again, indicated by the empty set.

```
hw:(pgm
|[]
|[]
[]
   (inj []
```

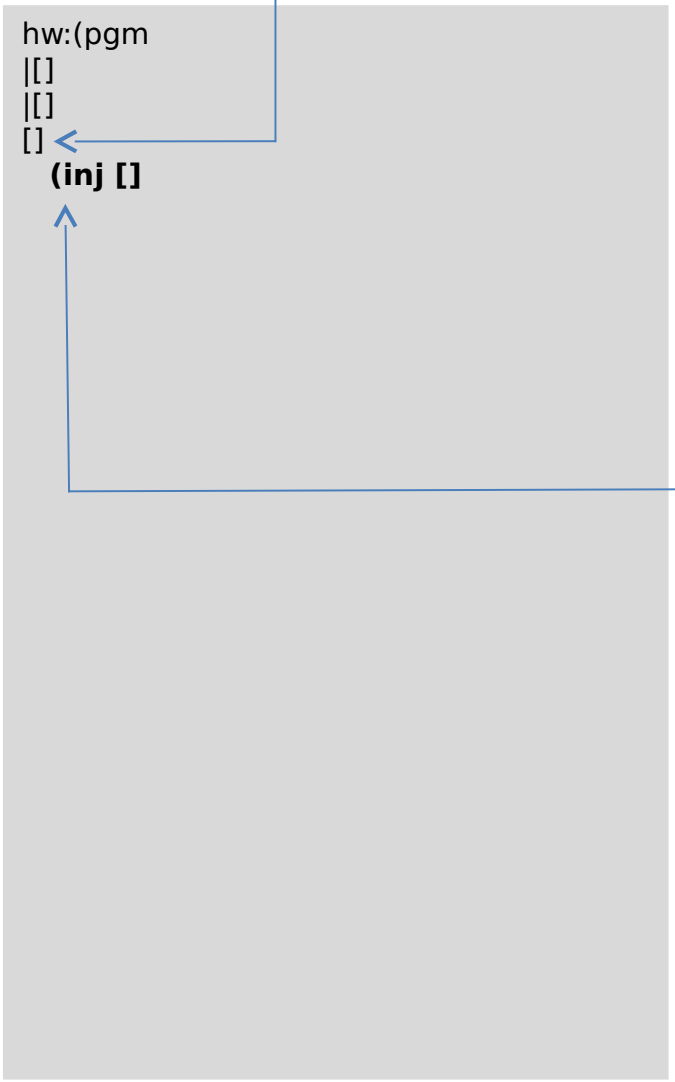
IMPORTANT:

In Replicode, you can specify populated sets in two ways:

1)[a b c]

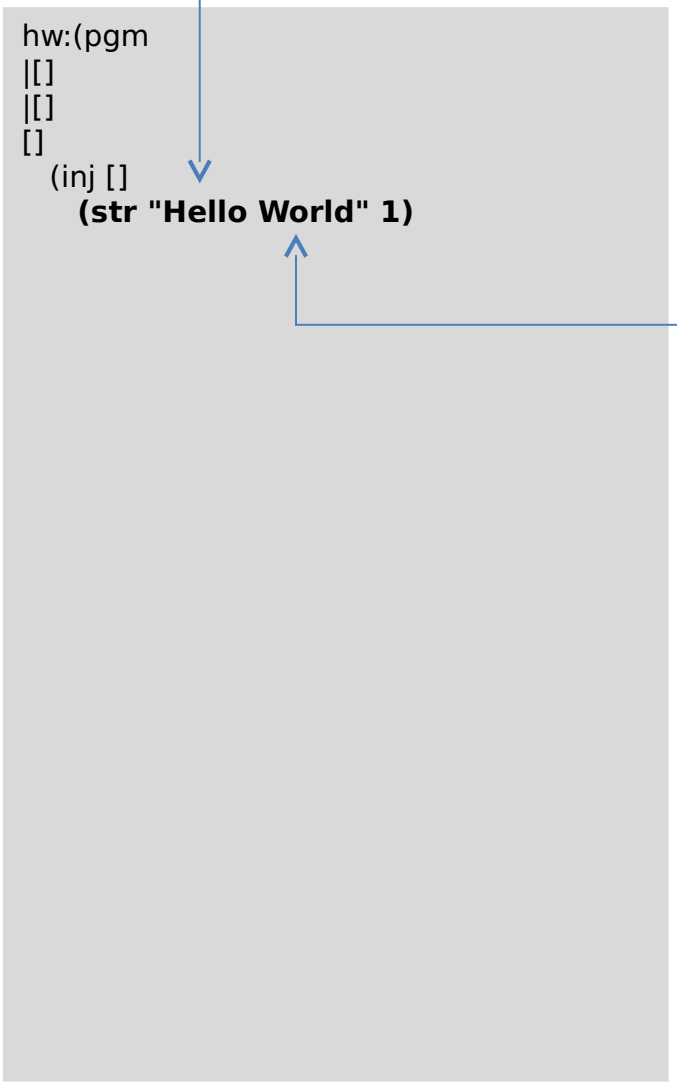
2) []
   a
   b
   c

The second method is used extensively as it improves readbility. It uses a **three space** indentation for elements of the set. The population of the set ends where the indentation stops.



The set of productions starts here. It is simply a set of operations that are executed when the program runs. Note that these operations will not occur unless input conditions are met (there are no input conditions in this example).

The beginning of an **injection** operation. An injection is basically adding an object into a particular group in the system. It can be a new object or an existing one.



This constructs a new **str** object. The class str is defined in the file **user.classes.replicode**. It has a single member of type st (string), whose value is set to „Hello World“.

This value is the threshold for propagation of saliency (psln_thr). This value controls how or if changes in the saliency of this object are spread to related objects. Not relevant in this example.

```
hw:(pgm
|[]
|[]
[]
   (inj []
      (str "Hello World" 1)
      [SYNC_FRONT now 1 forever root
nil]
```

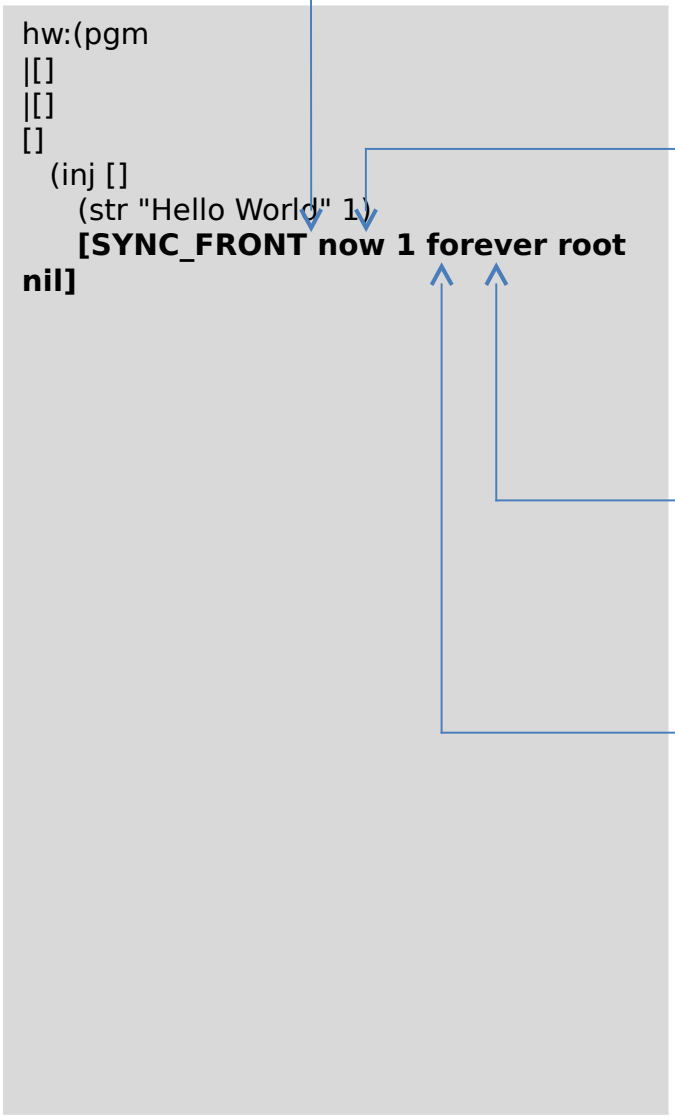
This set is called a **view**. It defines parameters for the injection, most importantly into which group we are injecting (root in this case). The view class (which this set is an instance of) is defined in the file **std.replicode**.

Injection time: Defines when we want the injection to occur. In this case we want it immediately so we use the **now** keyword. Otherwise, we can specify an absolute value in microseconds indicating a future point in time.

The **sync value** is either:

SYNC_BACK:
The object is always salient when its saliency value is above the groups saliency threshold

SYNC_FRONT:
The object is only salient when it crosses above the groups saliency threshold.



Saliency: This value specifies the initial saliency of the object. Saliency is a number between 0 and 1. The default saliency threshold of the root group is set to 0.5 so this object will immediately be eligable as input to programs.

Resilience: Specifies how long the object should exist, expressed as a multiple of the containg group's time scale, which is in turn expressed as a multiple of the base period. Keep in mind that programs can change the resilience of existing objects. In this case, the object should never be deleted so the keyword **forever** is used.

Origin: Specifies from what group or remote computing node this object came from. The keyword nil is used to skip the parameter.

Group: Specifies into which group this object should be injected.

IMPORTANT:
The same object can exist in multiple groups. However, it will always have a dedicated view for each group.

```
hw:(pgm
|[]
|[]
[]
  (inj []
    (str "Hello World" 1)
    [SYNC_FRONT now 1 forever root
nil]
  )
```

The injection command is now complete. Simply put, we are putting a new string object into the root group immediately. The string object has maximum saliency and will never be deleted.

```
hw:(pgm
|[]
|[]
[]
   (inj []
      (str "Hello World" 1)
      [SYNC_FRONT now 1 forever root nil]
   )
   (set [this.vw.act 0])
```

Set commands are used to change control parameters of objects (data and programs are both considered objects). Here we set the **activation** value of this instance of the program to zero. This disables the program, it will not run again. The default activation threshold of the root group is 0.4 (see the file **user.classes.replicode**).

this.vw.act is a reference to the current object:

this refers to the current object, the program hw in this case

vw refers to the view (of **this**)

act refers to the activation control

```
hw:(pgm
|[]
|[]
[]
   (inj []
      (str "Hello World" 1)
      [SYNC_FRONT now 1 forever root nil]
   )
   (set [this.vw.act 0])
1
)
|[]
```

Propagation of saliency threshold for this program.

Definition of the program ends.

Empty set for view. This is just a definition of the program and shall not belong to any particular group.

```
hw:(pgm
|[]
|[]
[]
   (inj []
      (str "Hello World" 1)
      [SYNC_FRONT now 1 forever root
nil]
   )
   (set [this.vw.act 0])
1
)
|[]
```

We now have a complete Replicode program definition. To make it run, we need to create an instance of it.

```
hw:(pgm
|[]
|[]
[]
   (inj []
      (str "Hello World" 1)
      [SYNC_FRONT now 1 forever root nil]
   )
   (set [this.vw.act 0])
1
)
|[]

ihw:(ipgm
```

Beginning of the command to create an instansiated program. The **ipgm** command creates an instance of a program that has already been defined, whereas the **pgm** command defines a program.

We label the instance in case we want to refer to it later.

```
hw:(pgm
|[]
|[]
[]
   (inj []
      (str "Hello World" 1)
      [SYNC_FRONT now 1 forever root nil]
   )
   (set [this.vw.act 0])
1
)
|[]

ihw:(ipgm
hw
|[]
```

Identify the program we want to instansiate. Corresponds to the label of the program definition above. This is why we labelled it.

This program does not require any parameters, so we use the empty set.

```
hw:(pgm
|[]
|[]
[]
   (inj []
      (str "Hello World" 1)
      [SYNC_FRONT now 1 forever root nil]
   )
   (set [this.vw.act 0])
1
)
|[]

ihw:(ipgm
hw
|[]
RUN_ALWAYS
1000000s
NOTIFY
```

Execution behavior:

RUN_ONCE: Instance will never execute more than once.
RUN_ALWAYS: Instance can execute multiple times. As the hw program requires no inputs, it would run continuously were it not for the **set** command that sets its activation to zero.

Time scope: Defines the maximum duration for an input pattern to fully occur.
Notification:

NOTIFY: Creates notifications when executed. Notifications are objects that are injected into the group of the program. These objects can be inputs for other programs.
SILENT: Never create notifications.

```
hw:(pgm
|[]
|[]
[]
   (inj []
      (str "Hello World" 1)
      [SYNC_FRONT now 1 forever root nil]
   )
   (set [this.vw.act 0])
1
)
|[]

ihw:(ipgm
hw
|[]
RUN_ALWAYS
100000us
NOTIFY
1
)
[]
   [SYNC_FRONT now 1 forever root
nil 1]
```

Propagation of saliency for this instance of the program.

The ipgm command ends.

Specify a view (already covered) for the instance. The ipgm command will inject the new instance according to this view.

```
hw:(pgm
|[]
|[]
[]
  (inj []
    (str "Hello World" 1)
    [SYNC_FRONT now 1 forever root
nil]
  )
  (set [this.vw.act 0])
1
)
|[]

ihw:(ipgm
hw
|[]
RUN_ALWAYS
100000us
NOTIFY
1
)
[]
  [SYNC_FRONT now 1 forever root
nil 1]
```

Our first program is now complete. Let‘s see what happens when we run it...

```
compiling ...
usr operators initialized
> User-defined operator library ./usr_operators.dll loaded
... done

Running for 1000 ms

Shutting rMem down...


DECOMPILATION

root:(grp 1 0.5 0.4 0 1 0 1 0 0 0 1 0 1 1 1 1 0.575 1 0 7.05297e-038
9.40395e-03
8 0 1 0 1 1 0 1 0 0 [nil] 1) []
   [true (_now) 0 forever nil nil false 0]

grp0:(grp 1 1 0 0 1 0 1 0 -0.2 0 0 0 1 2 1 1 0 0 0 9.40396e-038 9.40396e-038
1 1
 0 1 1 0 1 0 0 [nil] 1) []
   [true 0s:0ms:0us 0 forever root nil false 0]

stdin:(grp 1 0 0 0 1 0 1 0 0 1 0 0 1 1 1 1 0 0 1 0 0 1 1 0 1 1 0 1 0 0 [root] 1)
 []
   [true 0s:0ms:0us 0 forever grp0 nil true 1]
   [true 0s:0ms:0us 0 forever root nil false 0]

stdout:(grp 1 0 0 0 1 0 1 0 0 0 1 0 1 1 1 1 0 0 1 0 0 1 1 0 1 1 0 1 0 0 [nil] 1)
 []
   [true 0s:0ms:0us 0 forever root nil false 0]

self:(ent 0.95) []
   [true 0s:0ms:0us 0.6 forever root nil]

ent0:(ent 1) []
   [true 0s:0ms:0us 1 forever root nil]

pgm0:(pgm |[] |[] []
   (cmd _inj 0xa1000000 []
      (str "Hello World" 1)
      [true (_now) 1 forever root nil]
   )
   (cmd _set 0xa1000000 []
      this.vw.act
      0
   )
1) |[]

ipgm0:(ipgm pgm0 |[] true 0s:100ms:0us true 1) []
   [true 0s:0ms:0us 1 forever root nil 0]

str0:(str "Hello World" 1) []
   [true 0s:105ms:549us 1 forever root root]

mk.rdx0:(mk.rdx ipgm0 |[] [(cmd _inj 0xa1000000 []
   str0
   [true 0s:101ms:187us 1 forever root nil]
)] 1) []
   [true 0s:105ms:969us 1 992 root root]


Image taken at: Fri Mar 04 2011 14:19:23:542:396 GMT

10 objects
```

Our code is loaded and compiled. The Replicode executive runs for one second. Once it stops, the image (state of the system) is decompiled.

HÁSKÓLINN Í REYKJAVÍK
REYKJAVIK UNIVERSITY

```
compiling ...
usr operators initialized
> User-defined operator library ./usr_operators.dll loaded
... done

Running for 1000 ms

Shutting rMem down...


DECOMPILATION

root:(grp 1 0.5 0.4 0 1 0 1 0 0 0 1 0 1 1 1 1 0.575 1 0 7.05297e-038
9.40395e-03
8 0 1 0 1 1 0 1 0 0 [nil] 1) []
   [true (_now) 0 forever nil nil false 0]

grp0:(grp 1 1 0 0 1 0 1 0 -0.2 0 0 0 1 2 1 1 0 0 0 9.40396e-038 9.40396e-038
1 1
 0 1 1 0 1 0 0 [nil] 1) []
   [true 0s:0ms:0us 0 forever root nil false 0]

stdin:(grp 1 0 0 0 1 0 1 0 0 1 0 0 1 1 1 1 0 0 1 0 0 1 1 0 1 1 0 1 0 0 [root] 1)
 []
   [true 0s:0ms:0us 0 forever grp0 nil true 1]
   [true 0s:0ms:0us 0 forever root nil false 0]

stdout:(grp 1 0 0 0 1 0 1 0 0 0 1 0 1 1 1 1 0 0 1 0 0 1 1 0 1 1 0 1 0 0 [nil] 1)
 []
   [true 0s:0ms:0us 0 forever root nil false 0]

self:(ent 0.95) []
   [true 0s:0ms:0us 0.6 forever root nil]

ent0:(ent 1) []
   [true 0s:0ms:0us 1 forever root nil]

pgm0:(pgm |[] |[] []
   (cmd _inj 0xa1000000 []
      (str "Hello World" 1)
      [true (_now) 1 forever root nil]
   )
   (cmd _set 0xa1000000 []
      this.vw.act
      0
   )
1) |[]

ipgm0:(ipgm pgm0 |[] true 0s:100ms:0us true 1) []
   [true 0s:0ms:0us 1 forever root nil 0]

str0:(str "Hello World" 1) []
   [true 0s:105ms:549us 1 forever root root]

mk.rdx0:(mk.rdx ipgm0 |[] [(cmd _inj 0xa1000000 []
   str0
   [true 0s:101ms:187us 1 forever root nil]
)] 1) []
   [true 0s:105ms:969us 1 992 root root]


Image taken at: Fri Mar 04 2011 14:19:23:542:396 GMT

10 objects
```

Some standard groups and entities are created. This is defined in the file **user.classes.replicode**.

HÁSKÓLINN Í REYKJAVÍK
REYKJAVIK UNIVERSITY

```
compiling ...
usr operators initialized
> User-defined operator library ./usr_operators.dll loaded
... done

Running for 1000 ms

Shutting rMem down...


DECOMPILATION

root:(grp 1 0.5 0.4 0 1 0 1 0 0 0 1 0 1 1 1 1 0.575 1 0 7.05297e-038
9.40395e-03
8 0 1 0 1 1 0 1 0 0 [nil] 1) []
  [true (_now) 0 forever nil nil false 0]

grp0:(grp 1 1 0 0 1 0 1 0 -0.2 0 0 0 1 2 1 1 0 0 0 9.40396e-038 9.40396e-038
1 1
 0 1 1 0 1 0 0 [nil] 1) []
  [true 0s:0ms:0us 0 forever root nil false 0]

stdin:(grp 1 0 0 0 1 0 1 0 0 1 0 0 1 1 1 1 0 0 1 0 0 1 1 0 1 1 0 1 0 0 [root] 1)
 []
  [true 0s:0ms:0us 0 forever grp0 nil true 1]
  [true 0s:0ms:0us 0 forever root nil false 0]

stdout:(grp 1 0 0 0 1 0 1 0 0 0 1 0 1 1 1 1 0 0 1 0 0 1 1 0 1 1 0 1 0 0 [nil] 1)
 []
  [true 0s:0ms:0us 0 forever root nil false 0]

self:(ent 0.95) []
  [true 0s:0ms:0us 0.6 forever root nil]

ent0:(ent 1) []
  [true 0s:0ms:0us 1 forever root nil]

pgm0:(pgm |[] |[] []
  (cmd _inj 0xa1000000 []
    (str "Hello World" 1)
    [true (_now) 1 forever root nil]
  )
  (cmd _set 0xa1000000 []
    this.vw.act
    0
  )
1) |[]

ipgm0:(ipgm pgm0 |[] true 0s:100ms:0us true 1) []
  [true 0s:0ms:0us 1 forever root nil 0]

str0:(str "Hello World" 1) []
  [true 0s:105ms:549us 1 forever root root]

mk.rdx0:(mk.rdx ipgm0 |[] [(cmd _inj 0xa1000000 []
  str0
  [true 0s:101ms:187us 1 forever root nil]
)] 1) []
  [true 0s:105ms:969us 1 992 root root]


Image taken at: Fri Mar 04 2011 14:19:23:542:396 GMT

10 objects
```

Here is the definition for our program. Decompiled code is less readable than the original code.

HÁSKÓLINN Í REYKJAVÍK
REYKJAVIK UNIVERSITY

```
compiling ...
usr operators initialized
> User-defined operator library ./usr_operators.dll loaded
... done

Running for 1000 ms

Shutting rMem down...


DECOMPILATION

root:(grp 1 0.5 0.4 0 1 0 1 0 0 0 1 0 1 1 1 1 0.575 1 0 7.05297e-038
9.40395e-03
8 0 1 0 1 1 0 1 0 0 [nil] 1) []
   [true (_now) 0 forever nil nil false 0]

grp0:(grp 1 1 0 0 1 0 1 0 -0.2 0 0 0 1 2 1 1 0 0 0 9.40396e-038 9.40396e-038
1 1
 0 1 1 0 1 0 0 [nil] 1) []
   [true 0s:0ms:0us 0 forever root nil false 0]

stdin:(grp 1 0 0 0 1 0 1 0 0 1 0 0 1 1 1 1 0 0 1 0 0 1 1 0 1 1 0 1 0 0 [root] 1)
 []
   [true 0s:0ms:0us 0 forever grp0 nil true 1]
   [true 0s:0ms:0us 0 forever root nil false 0]

stdout:(grp 1 0 0 0 1 0 1 0 0 0 1 0 1 1 1 1 0 0 1 0 0 1 1 0 1 1 0 1 0 0 [nil] 1)
 []
   [true 0s:0ms:0us 0 forever root nil false 0]

self:(ent 0.95) []
   [true 0s:0ms:0us 0.6 forever root nil]

ent0:(ent 1) []
   [true 0s:0ms:0us 1 forever root nil]

pgm0:(pgm |[] |[] []
   (cmd _inj 0xa1000000 []
      (str "Hello World" 1)
      [true (_now) 1 forever root nil]
   )
   (cmd _set 0xa1000000 []
      this.vw.act
      0
   )
1) |[]

ipgm0:(ipgm pgm0 |[] true 0s:100ms:0us true 1) []
   [true 0s:0ms:0us 1 forever root nil 0]

str0:(str "Hello World" 1) []
   [true 0s:105ms:549us 1 forever root root]

mk.rdx0:(mk.rdx ipgm0 |[] [(cmd _inj 0xa1000000 []
   str0
   [true 0s:101ms:187us 1 forever root nil]
)] 1) []
   [true 0s:105ms:969us 1 992 root root]


Image taken at: Fri Mar 04 2011 14:19:23:542:396 GMT

10 objects
```

Here is the instance of the program we created with the **ipgm** command.

HÁSKÓLINN Í REYKJAVÍK
REYKJAVIK UNIVERSITY

```
compiling ...
usr operators initialized
> User-defined operator library ./usr_operators.dll loaded
... done

Running for 1000 ms

Shutting rMem down...


DECOMPILATION

root:(grp 1 0.5 0.4 0 1 0 1 0 0 0 1 0 1 1 1 1 0.575 1 0 7.05297e-038
9.40395e-03
8 0 1 0 1 1 0 1 0 0 [nil] 1) []
   [true (_now) 0 forever nil nil false 0]

grp0:(grp 1 1 0 0 1 0 1 0 -0.2 0 0 0 1 2 1 1 0 0 0 9.40396e-038 9.40396e-038
1 1
 0 1 1 0 1 0 0 [nil] 1) []
   [true 0s:0ms:0us 0 forever root nil false 0]

stdin:(grp 1 0 0 0 1 0 1 0 0 1 0 0 1 1 1 1 0 0 1 0 0 1 1 0 1 1 0 1 0 0 [root] 1)
 []
   [true 0s:0ms:0us 0 forever grp0 nil true 1]
   [true 0s:0ms:0us 0 forever root nil false 0]

stdout:(grp 1 0 0 0 1 0 1 0 0 0 1 0 1 1 1 1 0 0 1 0 0 1 1 0 1 1 0 1 0 0 [nil] 1)
 []
   [true 0s:0ms:0us 0 forever root nil false 0]

self:(ent 0.95) []
   [true 0s:0ms:0us 0.6 forever root nil]

ent0:(ent 1) []
   [true 0s:0ms:0us 1 forever root nil]

pgm0:(pgm |[] |[] []
   (cmd _inj 0xa1000000 []
      (str "Hello World" 1)
      [true (_now) 1 forever root nil]
   )
   (cmd _set 0xa1000000 []
      this.vw.act
      0
   )
1) |[]

ipgm0:(ipgm pgm0 |[] true 0s:100ms:0us true 1) []
   [true 0s:0ms:0us 1 forever root nil 0]

str0:(str "Hello World" 1) []
   [true 0s:105ms:549us 1 forever root root]

mk.rdx0:(mk.rdx ipgm0 |[] [(cmd _inj 0xa1000000 []
   str0
   [true 0s:101ms:187us 1 forever root nil]
)] 1) []
   [true 0s:105ms:969us 1 992 root root]


Image taken at: Fri Mar 04 2011 14:19:23:542:396 GMT

10 objects
```

This is the string object that the Hello World program injects. It was obviously executed since this exists.

HÁSKÓLINN Í REYKJAVÍK
REYKJAVIK UNIVERSITY

```
compiling ...
usr operators initialized
> User-defined operator library ./usr_operators.dll loaded
... done

Running for 1000 ms

Shutting rMem down...


DECOMPILATION

root:(grp 1 0.5 0.4 0 1 0 1 0 0 0 1 0 1 1 1 1 0.575 1 0 7.05297e-038
9.40395e-03
8 0 1 0 1 1 0 1 0 0 [nil] 1) []
  [true (_now) 0 forever nil nil false 0]

grp0:(grp 1 1 0 0 1 0 1 0 -0.2 0 0 0 1 2 1 1 0 0 0 9.40396e-038 9.40396e-038
1 1
 0 1 1 0 1 0 0 [nil] 1) []
  [true 0s:0ms:0us 0 forever root nil false 0]

stdin:(grp 1 0 0 0 1 0 1 0 0 1 0 0 1 1 1 1 0 0 1 0 0 1 1 0 1 1 0 1 0 0 [root] 1)
 []
  [true 0s:0ms:0us 0 forever grp0 nil true 1]
  [true 0s:0ms:0us 0 forever root nil false 0]

stdout:(grp 1 0 0 0 1 0 1 0 0 0 1 0 1 1 1 1 0 0 1 0 0 1 1 0 1 1 0 1 0 0 [nil] 1)
 []
  [true 0s:0ms:0us 0 forever root nil false 0]

self:(ent 0.95) []
  [true 0s:0ms:0us 0.6 forever root nil]

ent0:(ent 1) []
  [true 0s:0ms:0us 1 forever root nil]

pgm0:(pgm |[] |[] []
  (cmd _inj 0xa1000000 []
    (str "Hello World" 1)
    [true (_now) 1 forever root nil]
  )
  (cmd _set 0xa1000000 []
    this.vw.act
    0
  )
1) |[]

ipgm0:(ipgm pgm0 |[] true 0s:100ms:0us true 1) []
  [true 0s:0ms:0us 1 forever root nil 0]

str0:(str "Hello World" 1) []
  [true 0s:105ms:549us 1 forever root root]

mk.rdx0:(mk.rdx ipgm0 |[] [(cmd _inj 0xa1000000 [
  str0
  [true 0s:101ms:187us 1 forever root nil]
)] 1) []
  [true 0s:105ms:969us 1 992 root root]


Image taken at: Fri Mar 04 2011 14:19:23:542:396 GMT

10 objects
```

This is a notification from the program that it executed. It is called a **reduction marker**. It tells us that our instance injected a string objected and shows us the view that was used.

HÁSKÓLINN Í REYKJAVÍK
REYKJAVIK UNIVERSITY

In conclusion

- The Hello World example presents examples of basic operations in Replicode:
  Creating a program template
  - Instansiating a program
  - Creating an object (instance of a class)
  - Injecting programs and objects into groups
- The displayed output after execution is the decompiled system image
  - All objects that exist at the end of execution are shown
  - This includes any objects created (directly or indirectly) by the source code as well as notifications (reduction markers)
  - **Note**: Objects that were created and deleted before the end of execution are not shown