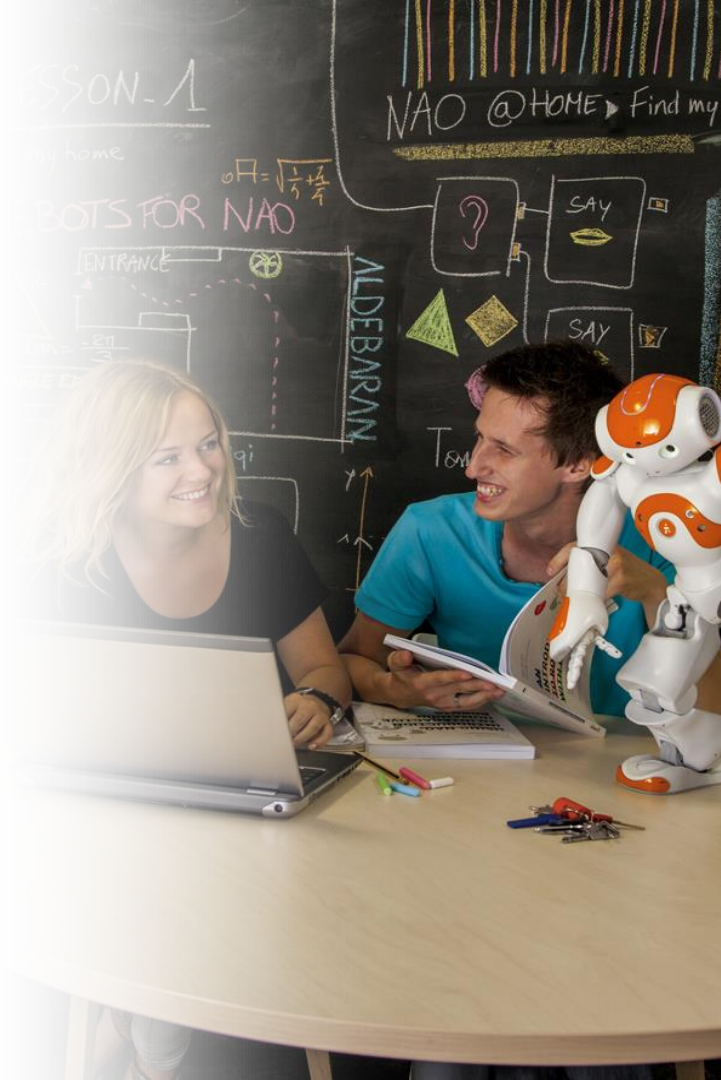
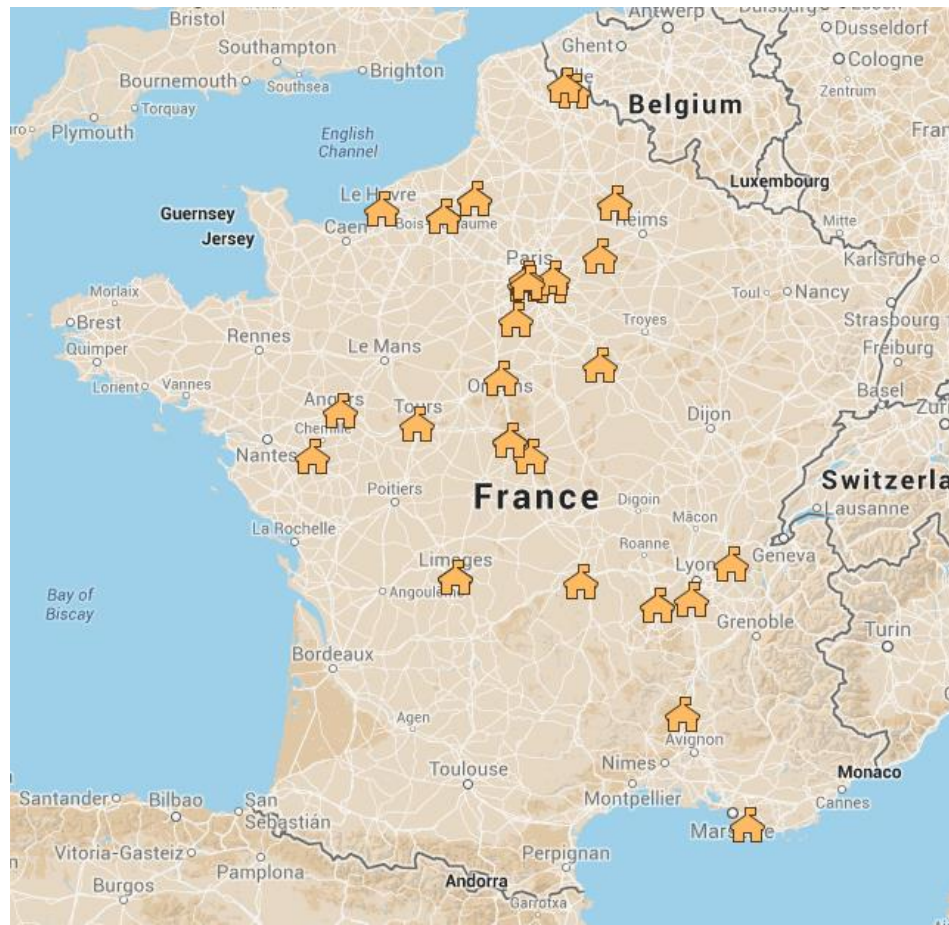


# NAO CHALLENGE @HOME

BIENVENUE !

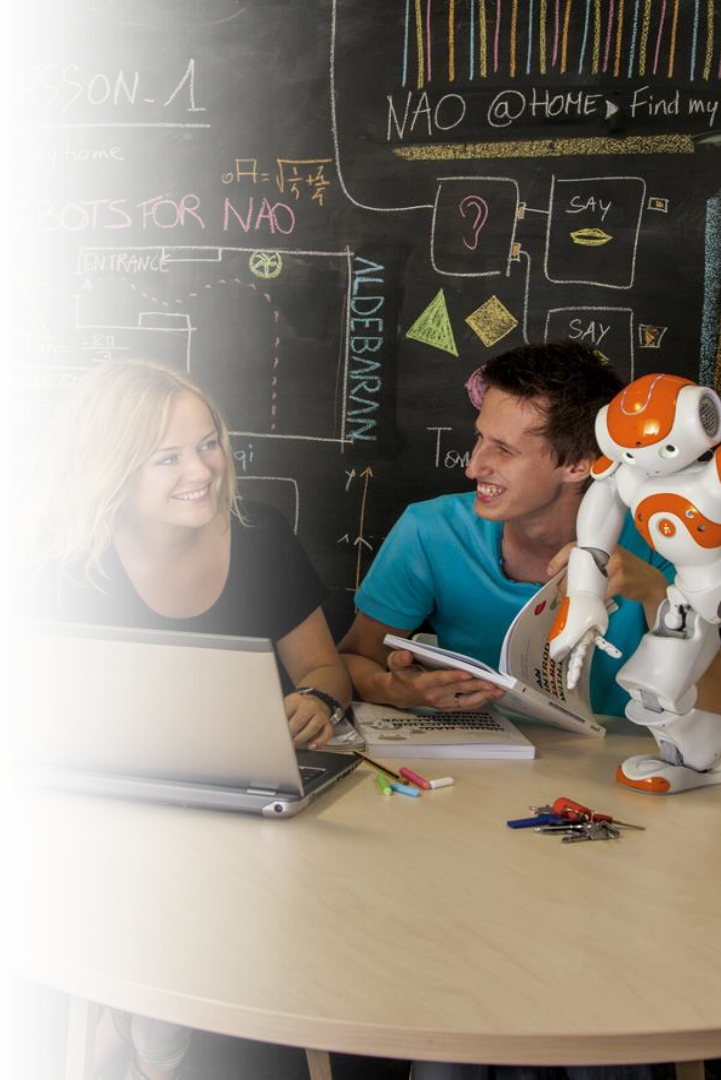


# BIENVENUE À TOUS !



# NAO CHALLENGE @HOME

C'EST PARTI !





# DEMANDEZ LE PROGRAMME !

## PREMIERS PAS AVEC NAO

- » Le robot
- » Choregraphe, Webots
- » Le SDK

## NAO CHALLENGE

- » Retour sur le règlement
- » Détails des épreuves

## EXEMPLES APPLIQUES

- » OpenCV
- » Création de mouvements

## CONCLUSION

- » Questions, Réponses !





# DEMANDEZ LE PROGRAMME !

## PREMIERS PAS AVEC NAO

- » Le robot
- » Choregraphe, Webots
- » Le SDK

## NAO CHALLENGE

- » Retour sur le règlement
- » Détails des épreuves

## EXEMPLES APPLIQUES

- » OpenCV
- » Création de mouvements

## CONCLUSION

- » Questions, Réponses !



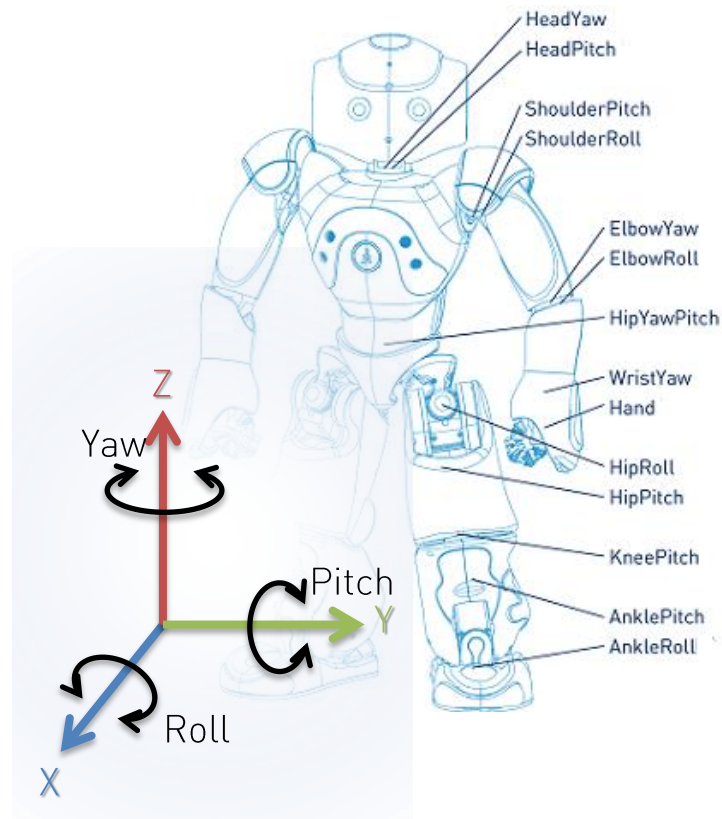
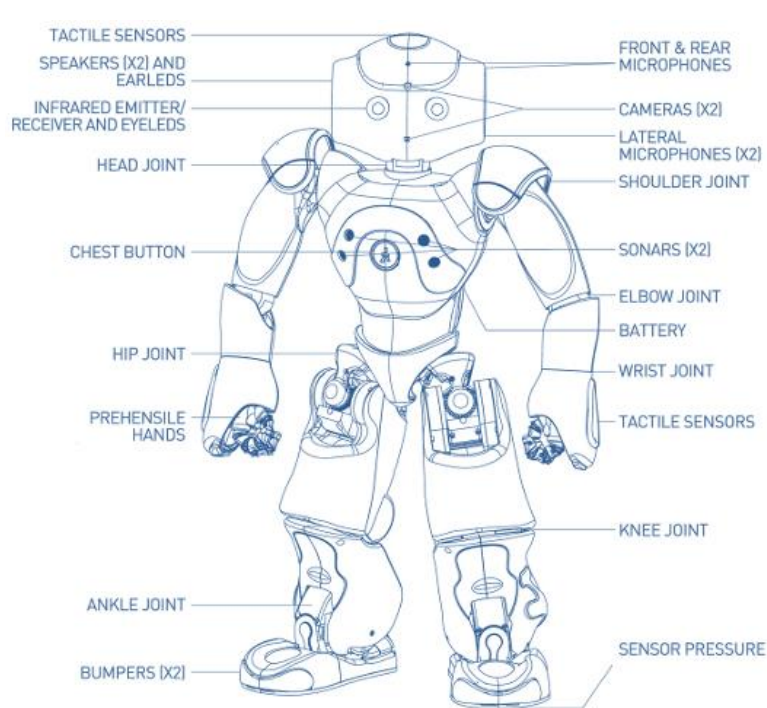
# CHECKLIST !

AUJOURD'HUI NOUS ALLONS UTILISER :

- » Webots 7
- » Choregraphe 1.14
- » Python 2.7
- » Numpy
- » Scipy
- » NAOqi Python SDK
- » OpenCV 2

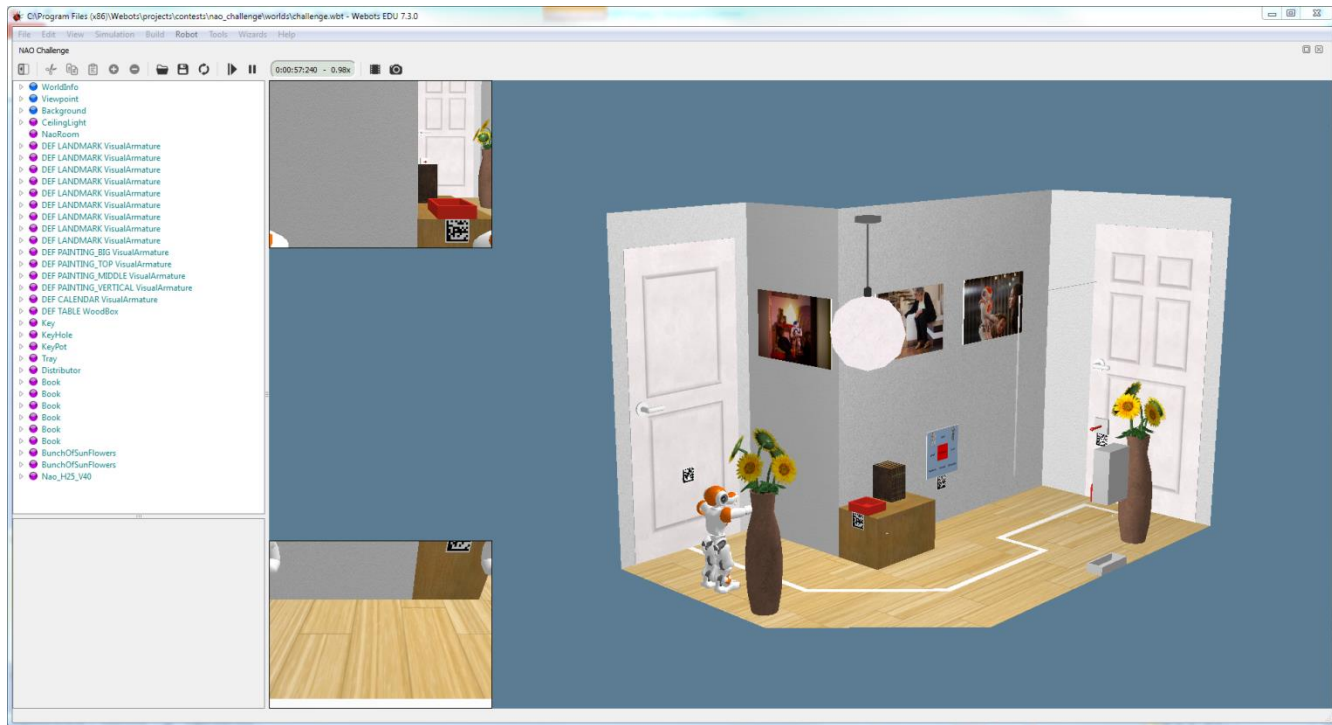


# LE ROBOT



# LOGICIELS

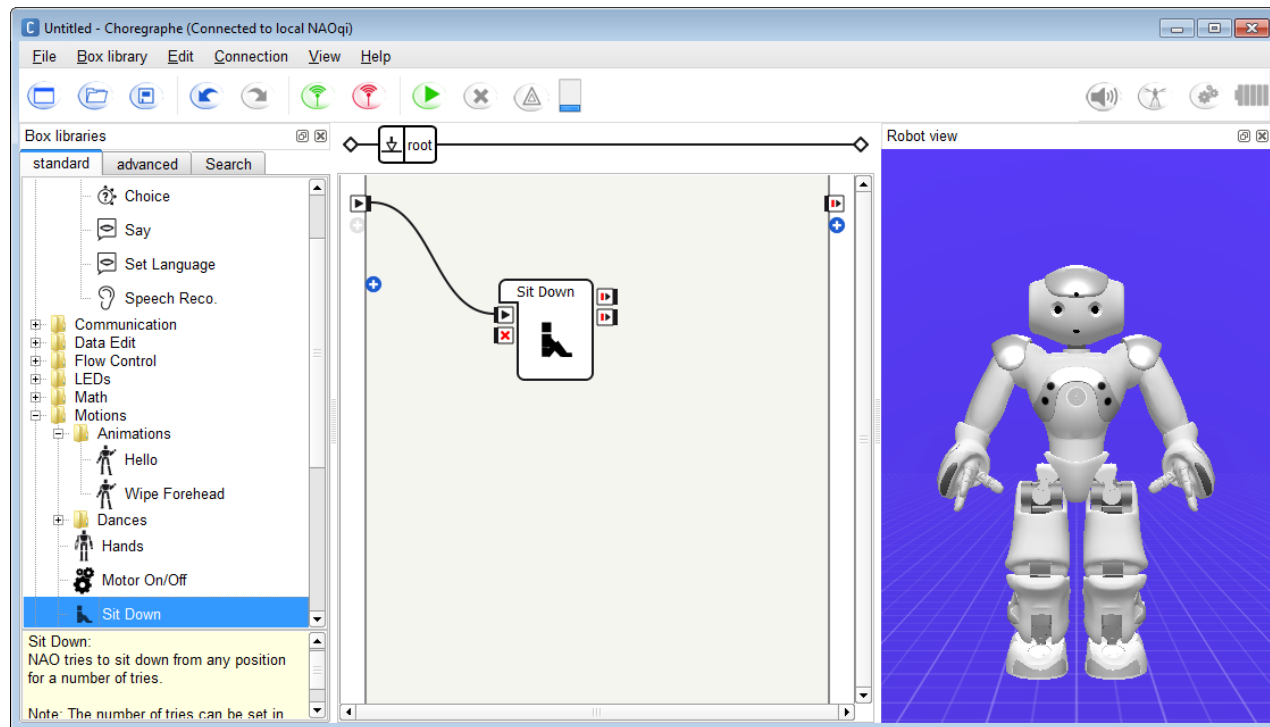
## WEBOTS





# LOGICIELS

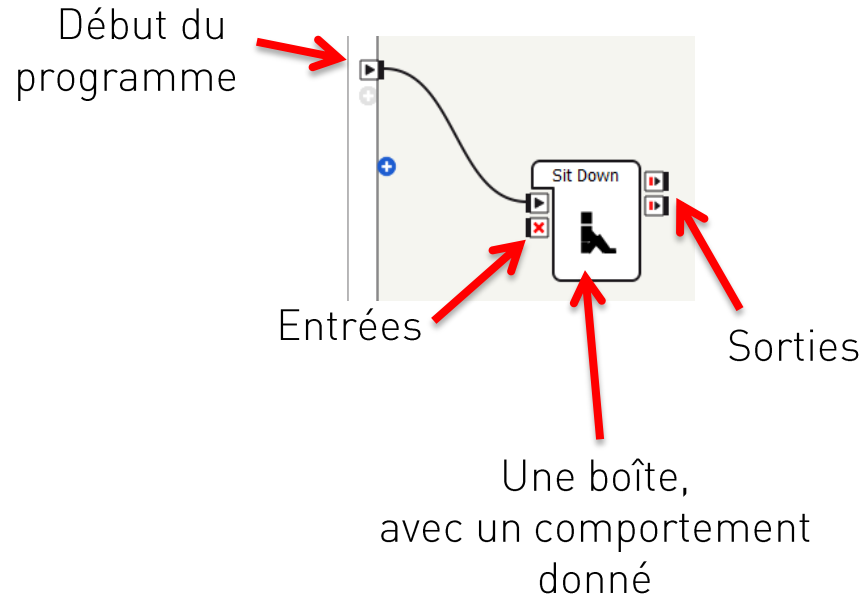
## CHOREGRAPHE



# LOGICIELS

## CHOREGRAPHE

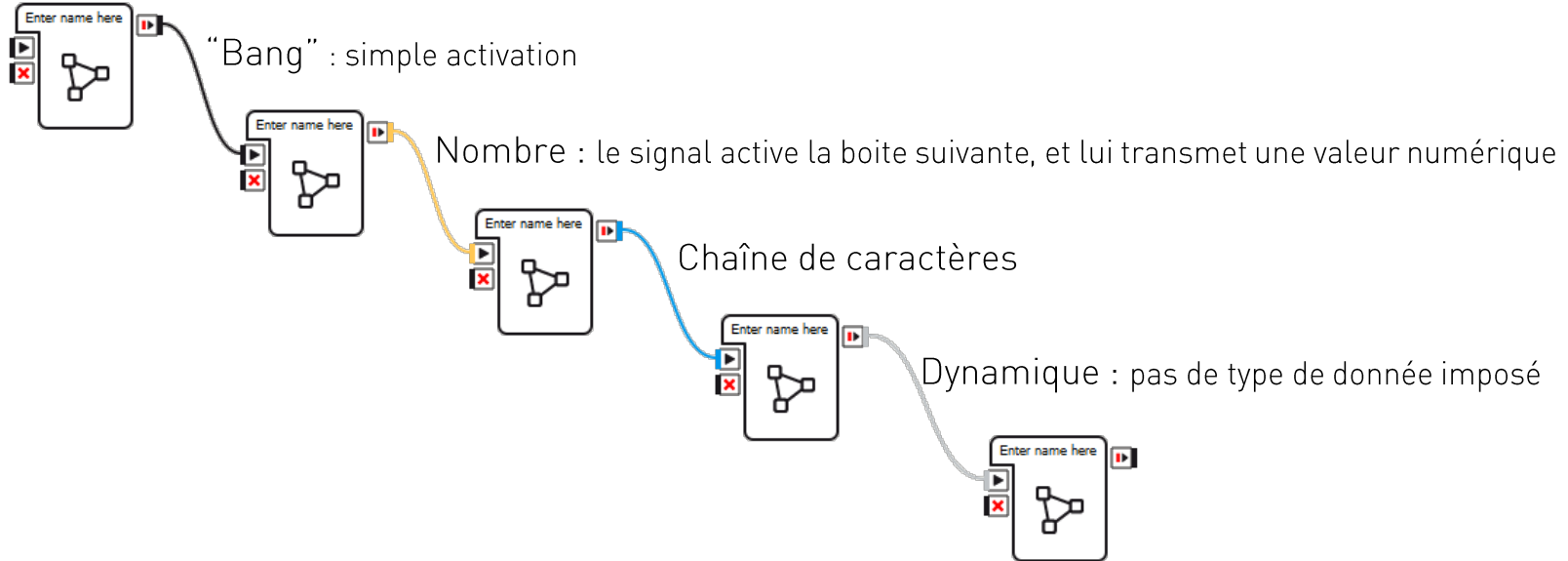
» Le principe : des boîtes, liées entre elles



# LOGICIELS

## CHOREGRAPHE

» Le principe : des boîtes, liées entre elles



# LOGICIELS

## LE SDK

- » Contrôler le robot avec les langages:
  - C++, Python
  - JAVA, Matlab, .NET, URBI, ROS, ...
- » Actionner les moteurs
- » Lire les valeurs de capteurs
- » Bibliothèque complète de fonctions

» Exemple en Python:

```
from naoqi import * ← Requis pour utiliser les fonctions de NAOqi  
  
tts = ALProxy( "ALTextToSpeech" , ip_du_robot , 9559 )  
  
tts.say( "Bonjour à tous, j'ai hate de travailler avec vous !" )
```

# LOGICIELS

## LE SDK

» Exemple en Python:

```
from naoqi import *
```

Connection au module ALTextToSpeech (Synthèse vocale) du robot :

```
tts = ALProxy( "ALTextToSpeech" , ip_du_robot , 9559 )
```

```
tts.say( "Bonjour à tous, j'ai hate de travailler avec vous !" )
```



# LOGICIELS

## LE SDK

» Exemple en Python:

```
from naoqi import *
```

Connection au module ALTextToSpeech (Synthèse vocale) du robot :

```
tts = ALProxy( "ALTextToSpeech" , ip_du_robot , 9559 )
```

```
tts.say( "Bonjour à tous, j'ai hate de travailler avec vous !" )
```

Utiliser un Proxy  
pour se connecter  
à un module

Nom du  
module

Adresse et port du robot  
Non nécessaire si  
embarqué sur le robot

» Exemple en Python:

```
from naoqi import *
```

Connection au module ALTextToSpeech (Synthèse vocale) du robot :

```
tts = ALProxy( "ALTextToSpeech" , ip_du_robot , 9559 )
```

```
tts.say( "Bonjour à tous, j'ai hate de travailler avec vous !" )
```

» Exemple en Python:

```
from naoqi import *
```

```
tts = ALProxy( "ALTextToSpeech" , ip_du_robot , 9559 )
```

Utilisation d'une fonction du module Text to Speech

```
tts.say( "Bonjour à tous, j'ai hate de travailler avec vous !" )
```

» Exemple en Python:

```
from naoqi import *
```

```
tts = ALProxy( "ALTextToSpeech" , ip_du_robot , 9559 )
```

Utilisation d'une fonction du module Text to Speech

```
tts.say( "Bonjour à tous, j'ai hate de travailler avec vous !" )
```

Le proxy créé  
plus haut

Fonction SAY  
du module

La phrase à dire

» Exemple en Python:

```
from naoqi import *
```

```
tts = ALProxy( "ALTextToSpeech" , ip_du_robot , 9559 )
```

Utilisation d'une fonction du module Text to Speech

```
tts.say( "Bonjour à tous, j'ai hate de travailler avec vous !" )
```

» Exemple en Python:

```
from naoqi import *
```

```
tts = ALProxy( "ALTextToSpeech" , ip_du_robot , 9559 )
```

```
tts.say( "Bonjour à tous, j'ai hate de travailler avec vous !" )
```



# LOGICIELS

## LE SDK

- » Où trouver les infos ?
  - Quels modules ? fonctions ? arguments ?
- » Documentation en ligne !

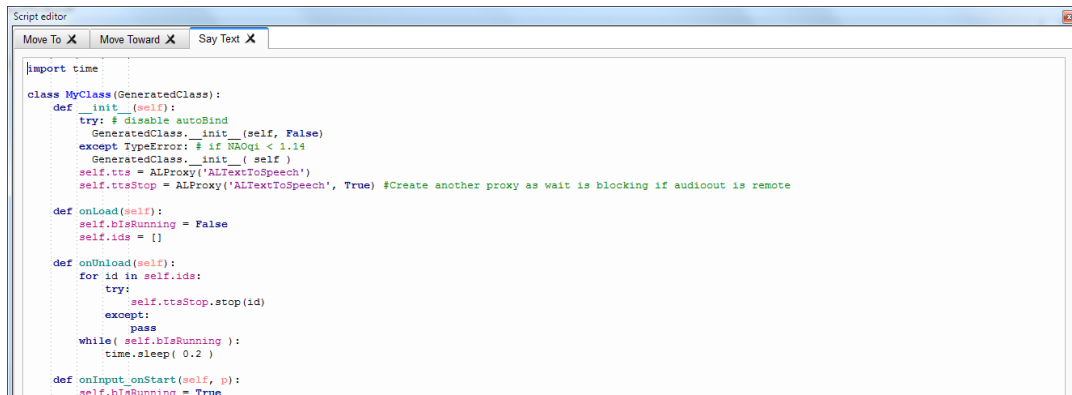


The screenshot shows the NAO Challenge website interface. At the top, there is a navigation bar with links: COMMUNITY, My NAO, PROJECTS, COMMUNITY, and RESOURCES. The 'RESOURCES' dropdown menu is open, showing options: SOFTWARE, DOCUMENTATION (highlighted with a red circle), TUTORIALS, and TRAININGS. Below the navigation bar, the main content area displays the 'NAO Challenge' header and a post titled 'Ressources pour le NAO Challenge 2013-2014'. The post includes metadata such as 'Views: 236', 'Downloads: 69', and 'Managers: nboudot, adesjobert'. The 'Description' section contains the text: 'Ressources pour le NAO Challenge 2013-2014'. Below this, there is a 'Forum' section with the text: 'Pour poser vos questions, partager vos expériences, et faire appel à toute la communauté de développement est dédié au NAO Challenge ici ! N'hésitez pas à aller vous présenter, présenter votre équipe !'. At the bottom, there is a link to 'Le règlement'.

# LOGICIELS

## LE SDK

- » Où trouver les infos ?
  - Quels modules ? fonctions ? arguments ?
- » Documentation en ligne !
- » Lisez le code des boîtes existantes !



```
Script editor
Move To X Move Toward X Say Text X

import time

class MyClass(GeneratedClass):
    def __init__(self):
        try: # disable autoBind
            GeneratedClass.__init__(self, False)
        except TypeError: # if NAOqi < 1.14
            GeneratedClass.__init__(self)
        self.tts = ALProxy('ALTextToSpeech')
        self.ttsStop = ALProxy('ALTextToSpeech', True) #Create another proxy as wait is blocking if audiostream is remote

    def onLoad(self):
        self.bIsRunning = False
        self.ids = []

    def onUnload(self):
        for id in self.ids:
            try:
                self.ttsStop.stop(id)
            except:
                pass
        while( self.bIsRunning ):
            time.sleep( 0.2 )

    def onInput_onStart(self, p):
        self.bIsRunning = True
```

# BASES DE PYTHON

## VARIABLES :

- » `A = 3`
- » `A = "bonjour"`

## LISTES :

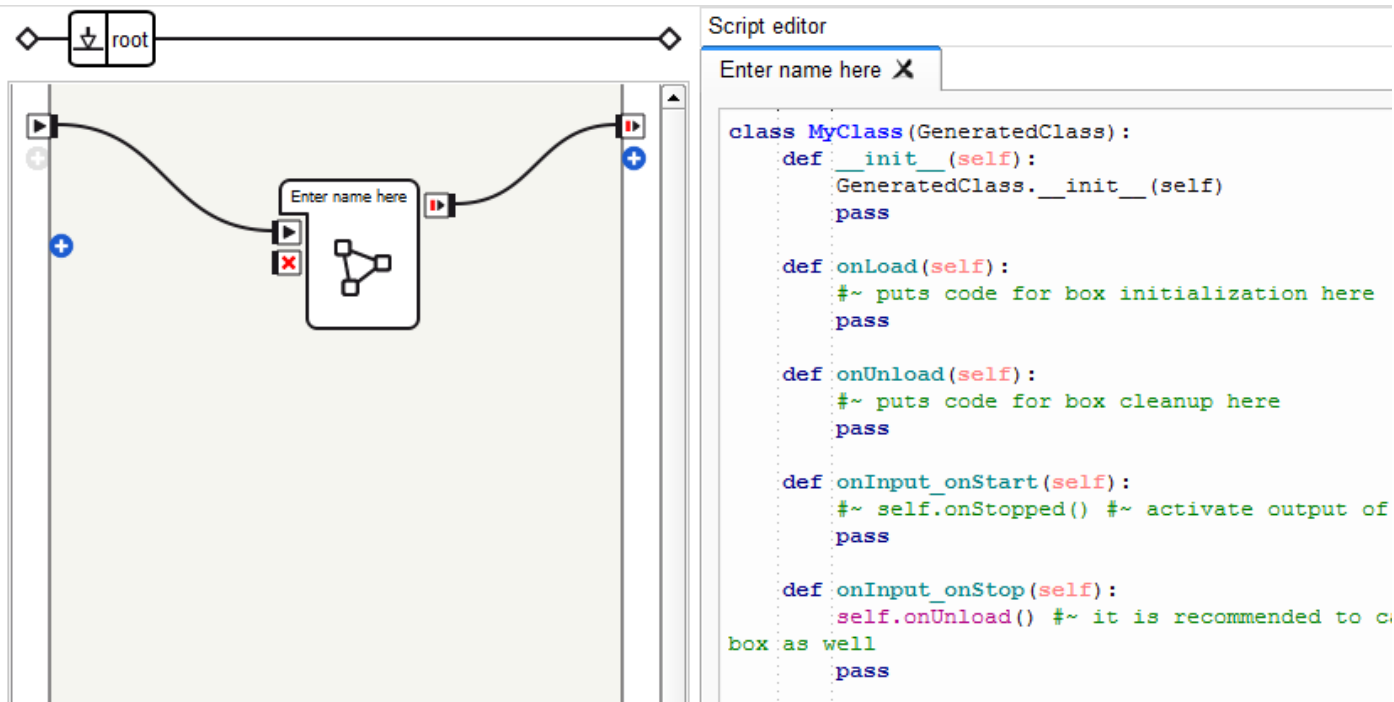
- » `A = [4, 3, 9, 8]`
- » `A[1] = 5` ➔ A devient `[4, 5, 9, 8]`
- » `A.append("Hello")` ➔ A devient `[4, 5, 9, 8, "hello"]`

# BASES DE PYTHON

## TESTS :

```
if (A == "truc"):  
    ce bloc fait quelque chose  
elif (A < 27):  
    ce bloc fait autre chose  
else:  
    ce bloc fait encore autre chose
```

- » Après if/while/for/def(...): Le bloc doit exister !
- » Pour un bloc qui ne fait rien : **"pass"**



The screenshot displays the Nao Choregraphe software interface. On the left, a sequence editor shows a 'root' node connected to a 'Enter name here' block, which contains a small graph icon. On the right, a 'Script editor' window is open, showing a Python class definition for 'MyClass' that inherits from 'GeneratedClass'.

```

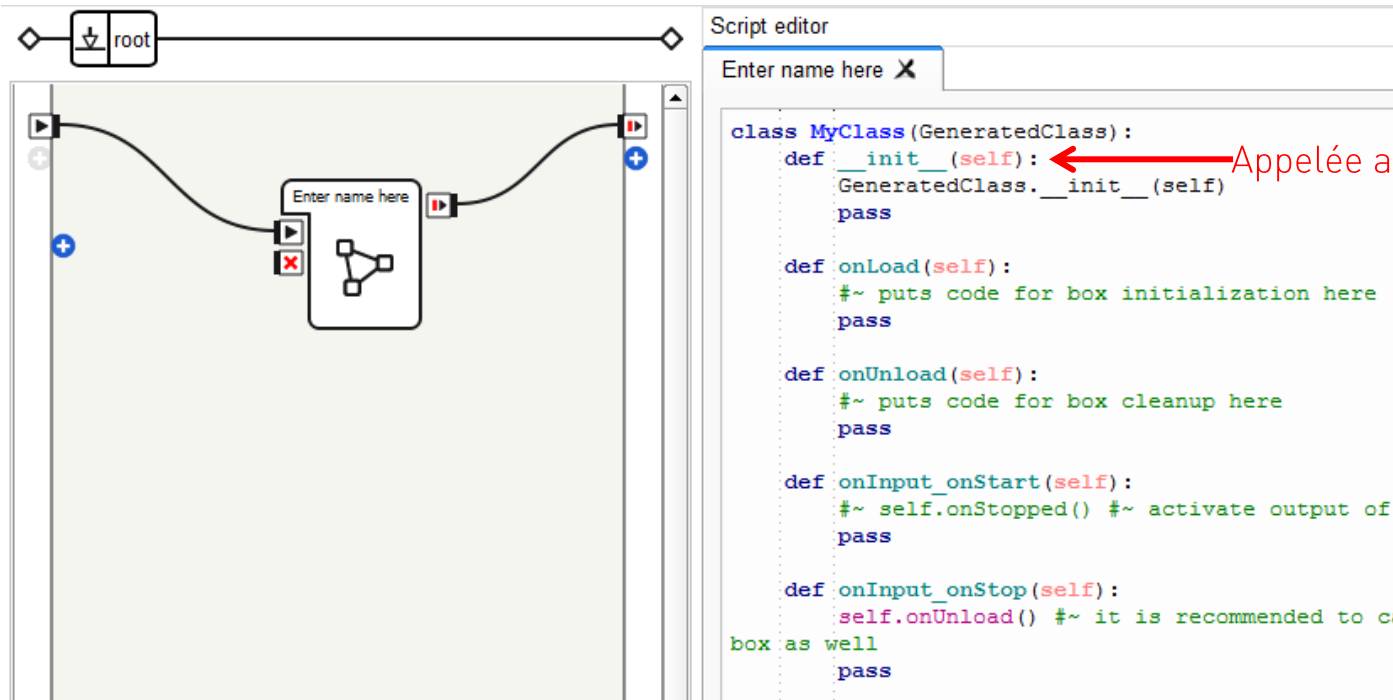
class MyClass(GeneratedClass):
    def __init__(self):
        GeneratedClass.__init__(self)
        pass

    def onLoad(self):
        #~ puts code for box initialization here
        pass

    def onUnload(self):
        #~ puts code for box cleanup here
        pass

    def onInput_onStart(self):
        #~ self.onStopped() #~ activate output of
        pass

    def onInput_onStop(self):
        self.onUnload() #~ it is recommended to c
        box as well
        pass
    
```



The screenshot shows the Nao Choregraphe interface. On the left, a choreography box labeled "Enter name here" contains a script editor. The script editor displays the following Python code:

```
class MyClass(GeneratedClass):
    def __init__(self):
        GeneratedClass.__init__(self)
        pass

    def onLoad(self):
        #~ puts code for box initialization here
        pass

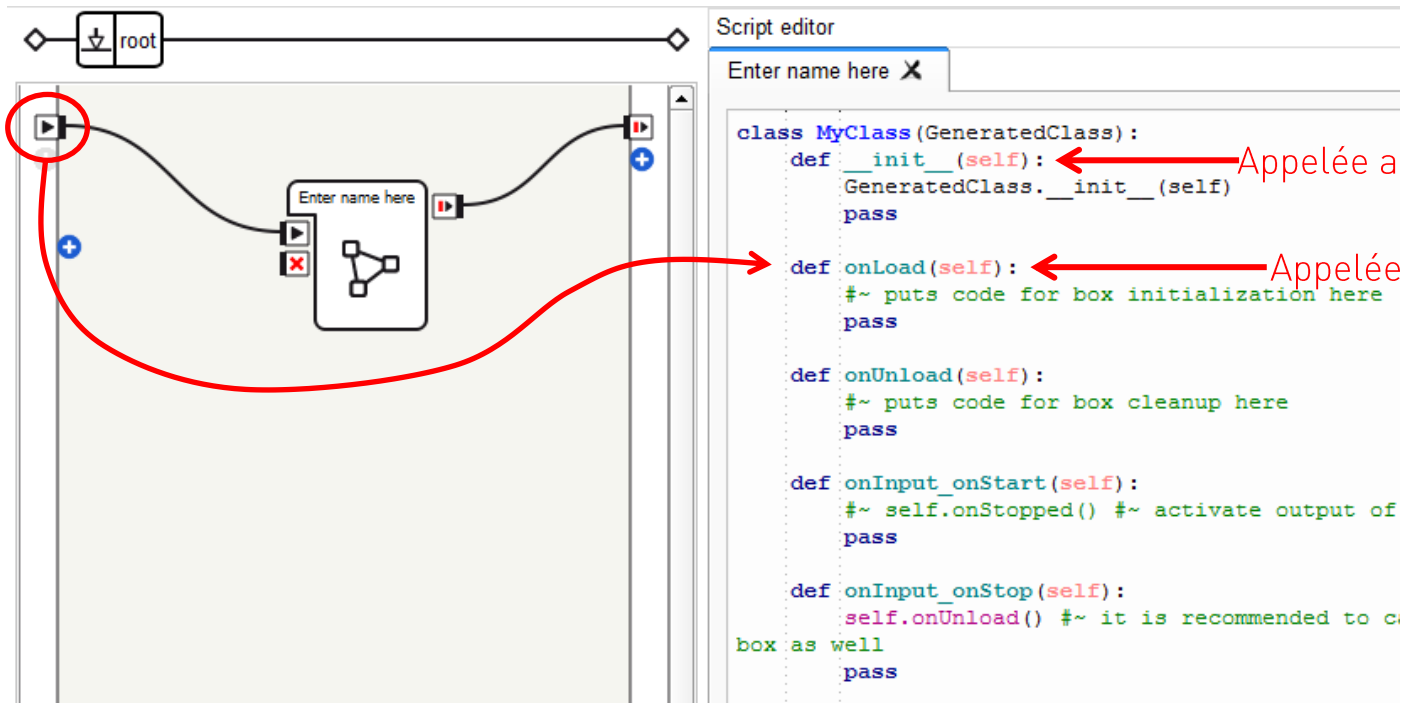
    def onUnload(self):
        #~ puts code for box cleanup here
        pass

    def onInput_onStart(self):
        #~ self.onStopped() #~ activate output of
        pass

    def onInput_onStop(self):
        self.onUnload() #~ it is recommended to c
        box as well
        pass
```

A red arrow points from the text "Appelée au lancement du programme" to the `__init__` method in the code.





The screenshot shows the Nao Choregraphe interface. On the left, a flow diagram is visible with a 'root' node and a box labeled 'Enter name here'. A red circle highlights the start button (a square with a play icon) in the flow diagram. A red arrow points from this button to the 'onLoad' method in the script editor on the right. Another red arrow points from the 'Enter name here' box to the 'onLoad' method. The script editor shows the following code:

```
class MyClass(GeneratedClass):
    def __init__(self):
        GeneratedClass.__init__(self)
        pass

    def onLoad(self):
        #~ puts code for box initialization here
        pass

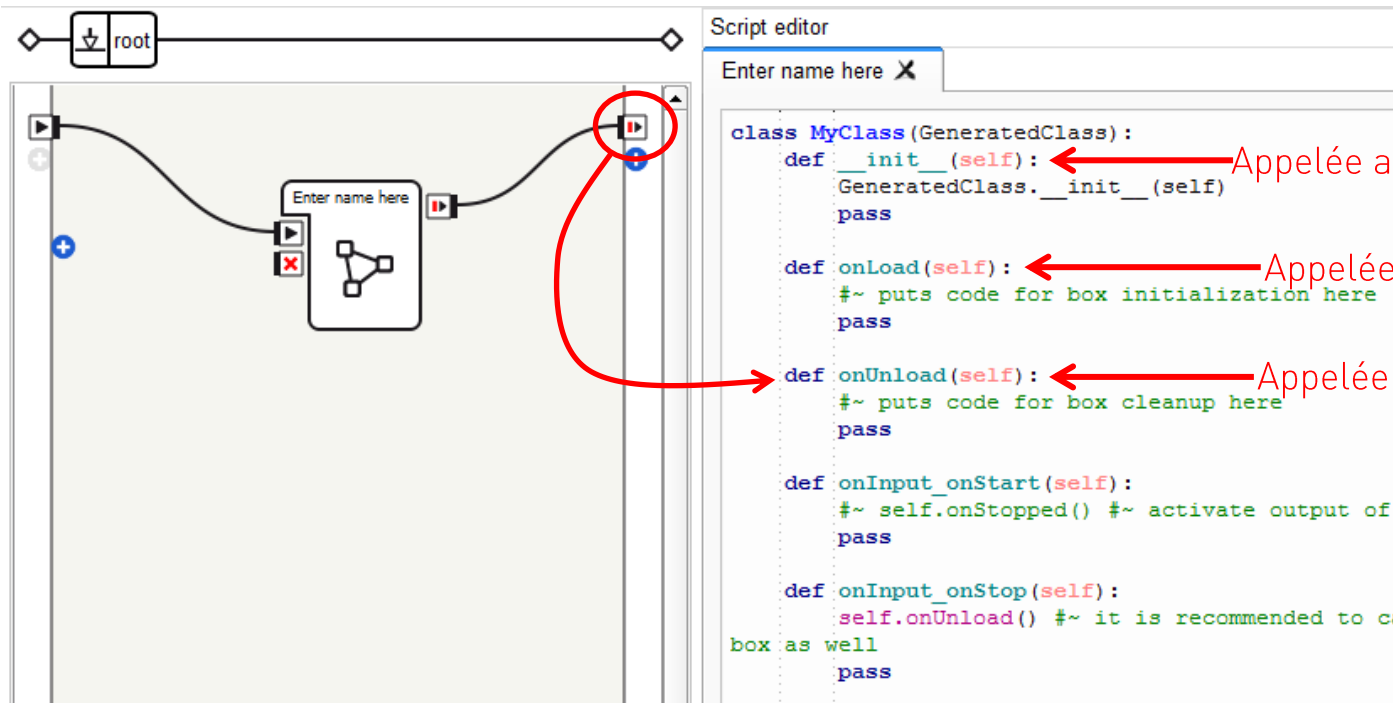
    def onUnload(self):
        #~ puts code for box cleanup here
        pass

    def onInput_onStart(self):
        #~ self.onStopped() #~ activate output of
        pass

    def onInput_onStop(self):
        self.onUnload() #~ it is recommended to c
        box as well
        pass
```

Appelée au lancement du programme

Appelée à l'entrée du Flow Diagram



Script editor

Enter name here X

```
class MyClass(GeneratedClass):
    def __init__(self):
        GeneratedClass.__init__(self)
        pass

    def onLoad(self):
        #~ puts code for box initialization here
        pass

    def onUnload(self):
        #~ puts code for box cleanup here
        pass

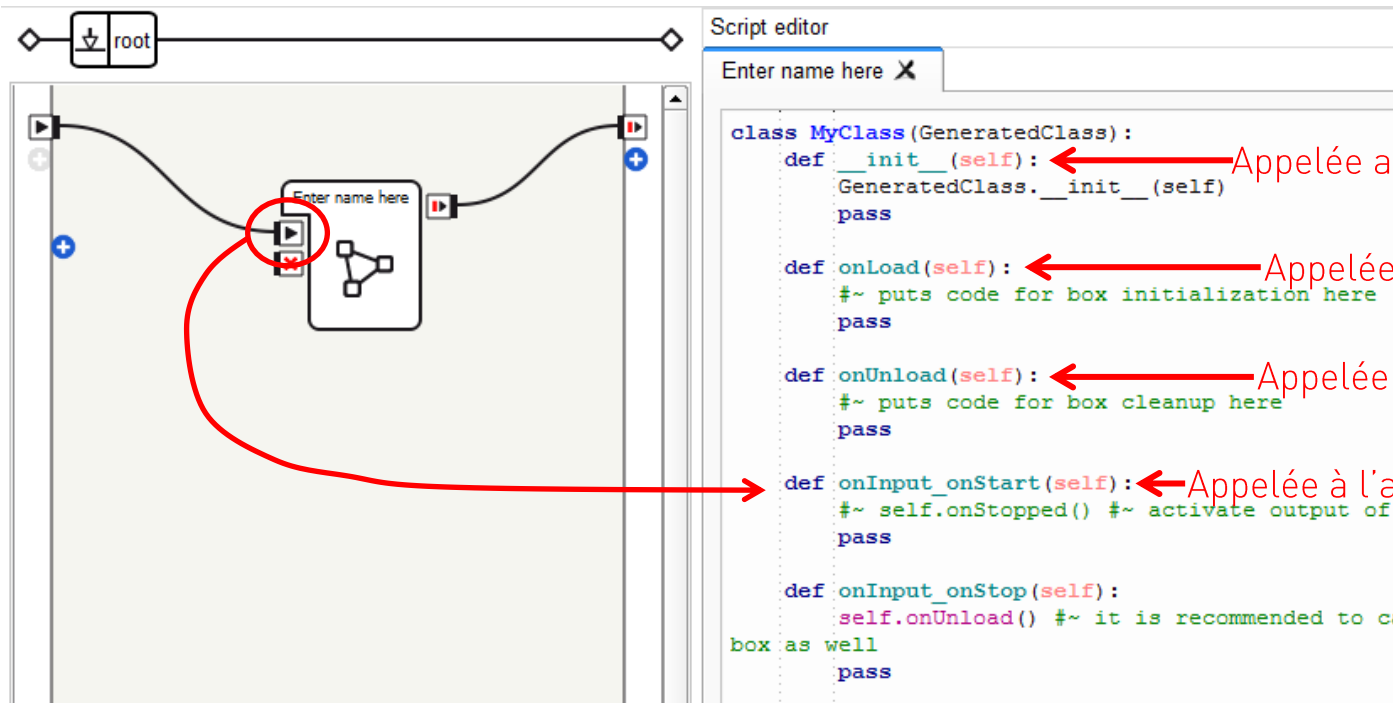
    def onInput_onStart(self):
        #~ self.onStopped() #~ activate output of
        pass

    def onInput_onStop(self):
        self.onUnload() #~ it is recommended to c
        box as well
        pass
```

Appelée au lancement du programme

Appelée à l'entrée du Flow Diagram

Appelée à la sortie du Flow Diagram



The image shows the NAO Choregraphe interface. On the left is a flow diagram with a 'root' node and a box labeled 'Enter name here'. A red circle highlights this box, and a red arrow points from it to the 'onInput\_onStart' method in the script editor. The script editor on the right contains the following Python code:

```
class MyClass(GeneratedClass):
    def __init__(self):
        GeneratedClass.__init__(self)
        pass

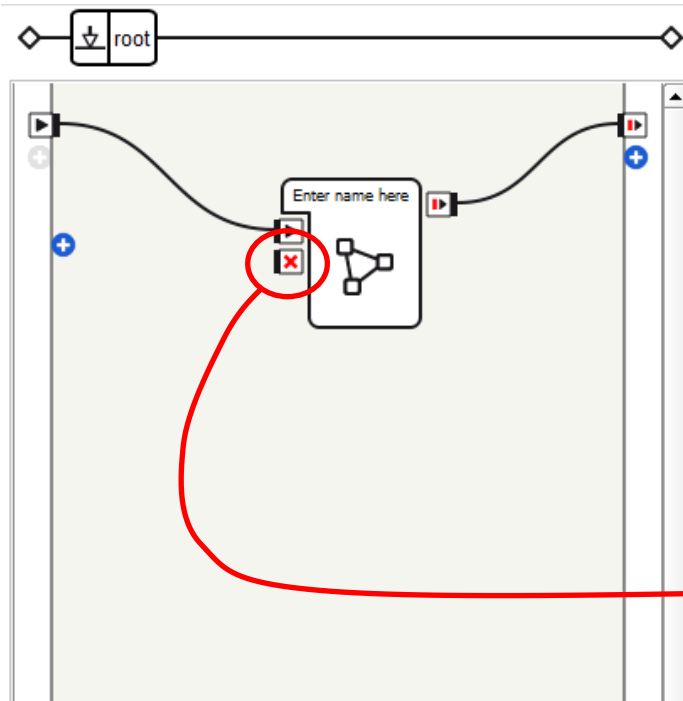
    def onLoad(self):
        #~ puts code for box initialization here
        pass

    def onUnload(self):
        #~ puts code for box cleanup here
        pass

    def onInput_onStart(self):
        #~ self.onStopped() #~ activate output of
        pass

    def onInput_onStop(self):
        self.onUnload() #~ it is recommended to c
        box as well
        pass
```

Red arrows indicate the execution flow: from the 'onInput\_onStart' method to the 'Enter name here' box, and from the 'onLoad' and 'onUnload' methods to the flow diagram.



Script editor

Enter name here X

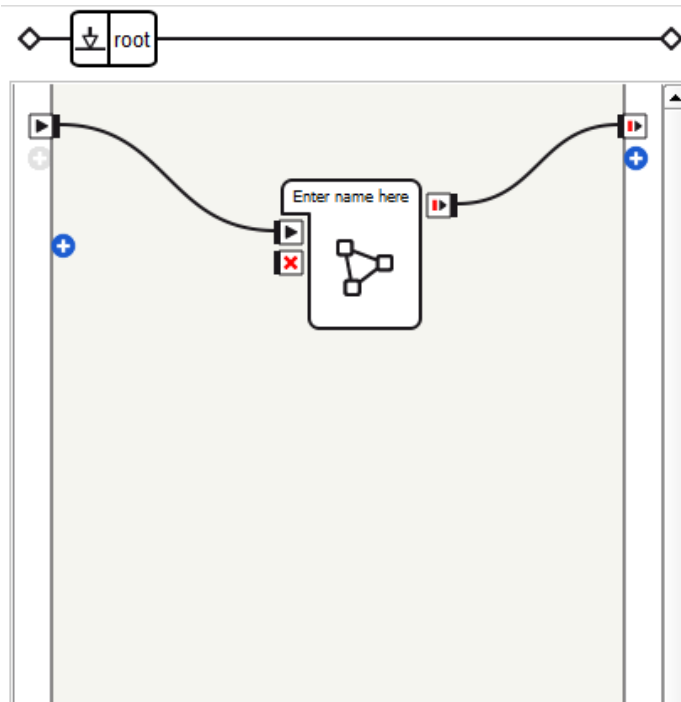
```
class MyClass(GeneratedClass):
    def __init__(self): ← Appelée au lancement du programme
        GeneratedClass.__init__(self)
        pass

    def onLoad(self): ← Appelée à l'entrée du Flow Diagram
        #~ puts code for box initialization here
        pass

    def onUnload(self): ← Appelée à la sortie du Flow Diagram
        #~ puts code for box cleanup here
        pass

    def onInput_onStart(self): ← Appelée à l'activation de l'entrée onStart
        #~ self.onStopped() #~ activate output of
        pass

    def onInput_onStop(self): ← Appelée à l'activation de l'entrée onStop
        self.onUnload() #~ it is recommended to c
        box as well
        pass
```



Script editor

Enter name here X

```
class MyClass(GeneratedClass):
    def __init__(self): ← Appelée au lancement du programme
        GeneratedClass.__init__(self)
        pass

    def onLoad(self): ← Appelée à l'entrée du Flow Diagram
        #~ puts code for box initialization here
        pass

    def onUnload(self): ← Appelée à la sortie du Flow Diagram
        #~ puts code for box cleanup here
        pass

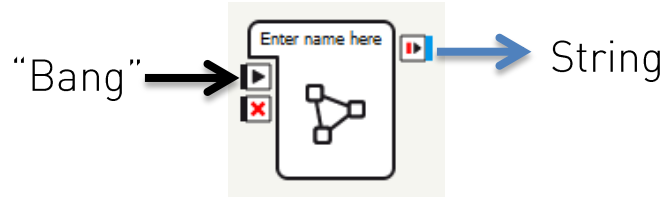
    def onInput_onStart(self): ← Appelée à l'activation de l'entrée onStart
        #~ self.onStopped() #~ activate output of
        pass

    def onInput_onStop(self): ← Appelée à l'activation de l'entrée onStop
        self.onUnload() #~ it is recommended to c
        pass

box as well
pass
```

# BASES DE PYTHON

## DANS UNE BOITE DE CHOREGRAPHE :



» Lire le contenu d'un signal :

```
def onInput_onStart(self, valeur):  
    maVariable = valeur
```

Un nouvel argument !

» Envoyer une valeur:

```
self.onStopped(valeur)
```

Le nom de la sortie

Valeur à envoyer



# BASES DE PYTHON

## UN PREMIER EXERCICE !

- » Créez une boîte « Ecrire dans la mémoire »
  - Type : Script
  - Entrée : Type « Dynamic », Nature « onStart »
  - Sortie : Type « Bang », Nature « onStoped »
  - Paramètre : « Variable », type « String », valeur par défaut « maVariable »
  - À l'initialisation : Connexion à ALMemory
  - À l'activation de l'entrée : Enregistrer le contenu du signal d'entrée dans la mémoire, sous le nom entré en paramètre.



- » Créez une boîte « Lire dans la mémoire »
  - Type : Script
  - Entrée : Type « Bang », nature « onStart »
  - Sortie : Type « Dynamic », nature « onStoped »
  - Paramètre : « Variable », type « String », valeur par défaut « maVariable »
  - À l'initialisation : Connexion à ALMemory
  - À l'activation de l'entrée : Lire dans la mémoire la variable dont le nom est entré par l'utilisateur en paramètre.



# PREMIERS PAS

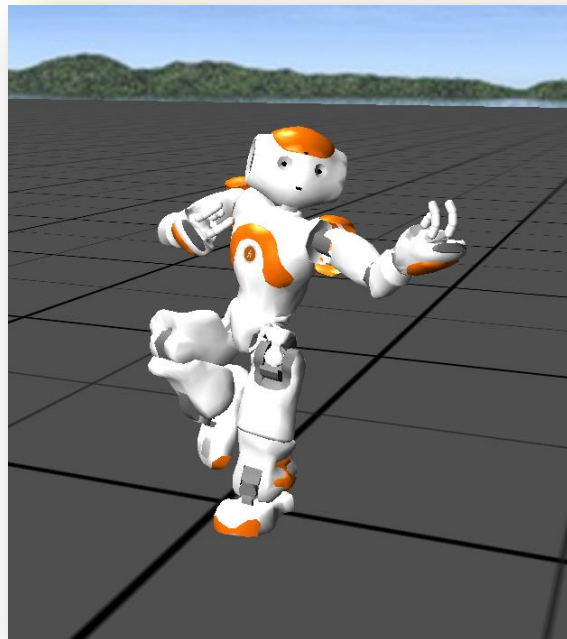
Choregraphe

NAO

QUESTIONS ?

Webots

SDK





# DEMANDEZ LE PROGRAMME !

## PREMIERS PAS AVEC NAO

- » Le robot
- » Choregraphe, Webots
- » Le SDK

## NAO CHALLENGE

- » Retour sur le règlement
- » Détails des épreuves

## EXEMPLES APPLIQUES

- » OpenCV
- » Création de mouvements

## CONCLUSION

- » Questions, Réponses !



# DEMANDEZ LE PROGRAMME !

## PREMIERS PAS AVEC NAO

- » Le robot
- » Choregraphe, Webots
- » Le SDK

## NAO CHALLENGE

- » Retour sur le règlement
- » Détails des épreuves

## EXEMPLES APPLIQUES

- » OpenCV
- » Création de mouvements

## CONCLUSION

- » Questions, Réponses !





# NAO CHALLENGE

## LES EPREUVES

- » Clé
- » Calendrier
- » Distributeur

## DOCUMENTATION

- » Community
- » Forum

# NAO CHALLENGE

## LES EPREUVES

- » Clé
- » Calendrier
- » Distributeur

## DOCUMENTATION

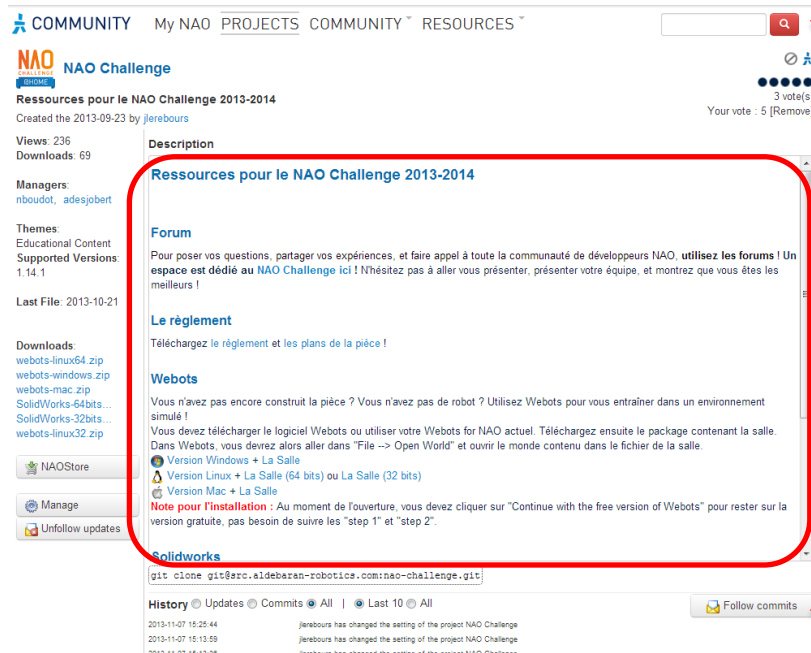
- » Community
- » Forum



# NAO CHALLENGE

## DOCUMENTATION

### » Community : projet NAO Challenge



COMMUNITY My NAO PROJECTS COMMUNITY RESOURCES

**NAO Challenge**

Ressources pour le NAO Challenge 2013-2014

Created the 2013-09-23 by jerebours

Views: 236  
Downloads: 69

Managers:  
nboudst, adesjobert

Themes:  
Educational Content  
Supported Versions:  
1.14.1

Last File: 2013-10-21

Downloads:  
webots-linux64.zip  
webots-windows.zip  
webots-mac.zip  
SolidWorks-64bits...  
SolidWorks-32bits...  
webots-linux32.zip

NAOStore

Manage

Unfollow updates

**Description**

**Ressources pour le NAO Challenge 2013-2014**

**Forum**

Pour poser vos questions, partager vos expériences, et faire appel à toute la communauté de développeurs NAO, **utilisez les forums** ! Un espace est dédié au **NAO Challenge** ici ! N'hésitez pas à aller vous présenter, présenter votre équipe, et montrez que vous êtes les meilleurs !

**Le règlement**

Téléchargez le règlement et les plans de la pièce !

**Webots**

Vous n'avez pas encore construit la pièce ? Vous n'avez pas de robot ? Utilisez Webots pour vous entraîner dans un environnement simulé !

Vous devez télécharger le logiciel Webots ou utiliser votre Webots pour NAO actuel. Téléchargez ensuite le package contenant la salle. Dans Webots, vous devez alors aller dans "File -> Open World" et ouvrir le monde contenu dans le fichier de la salle.

Version Windows + La Salle  
Version Linux + La Salle (64 bits) ou La Salle (32 bits)  
Version Mac + La Salle

**Note pour l'installation** : Au moment de l'ouverture, vous devez cliquer sur "Continue with the free version of Webots" pour rester sur la version gratuite, pas besoin de suivre les "step 1" et "step 2".

**Solidworks**

git clone git@arc.aldebaran-robotics.com:nao-challenge.git

**History** Updates Commits All Last 10 All

2013-11-07 15:25:44 jerebours has changed the setting of the project NAO Challenge  
2013-11-07 15:13:59 jerebours has changed the setting of the project NAO Challenge  
2013-11-07 15:13:59 jerebours has changed the setting of the project NAO Challenge

Follow commits

Champ de recherche pour retrouver le projet (ou en chercher d'autres!)

Liens intéressants pour le projet

"Follow Commits" pour recevoir un mail à chaque mise à jour de cette page

# NAO CHALLENGE

## DOCUMENTATION

### » Forum NAO Challenge

COMMUNITY My NAO PROJECTS COMMUNITY RESOURCES

Forum » NAO Challenge (3 threads, 6 posts)

	Thread	Posts	Views	Subscribed	Last Post
	<a href="#">Bienvenue !</a>	1	40	4	2 weeks, 3 days ago by jlerebours <a href="#">[view]</a>
	<a href="#">Présentez-vous</a>	2	40	5	20 h, 39 min ago by lboutin <a href="#">[view]</a>
	<a href="#">Mises à jour et Précisions sur le règlement</a>	3	60	4	23 h, 19 min ago by jlerebours <a href="#">[view]</a>

Unsubscribe from this forum
 Update Subscriptions
 Create a new thread

Toutes les informations importantes seront transmises par ce forum !

Pour recevoir un mail à chaque nouveau message

Pour créer une nouvelle discussion visible par tous





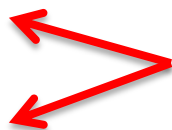
# NAO CHALLENGE

## LES EPREUVES

- » Clé
- » Calendrier
- » Distributeur

## DOCUMENTATION

- » Community
- » Forum



Inscrivez-vous et participez !



# NAO CHALLENGE

## LES EPREUVES

- » Clé
- » Calendrier
- » Distributeur

Les 3 épreuves sont  
séparées !

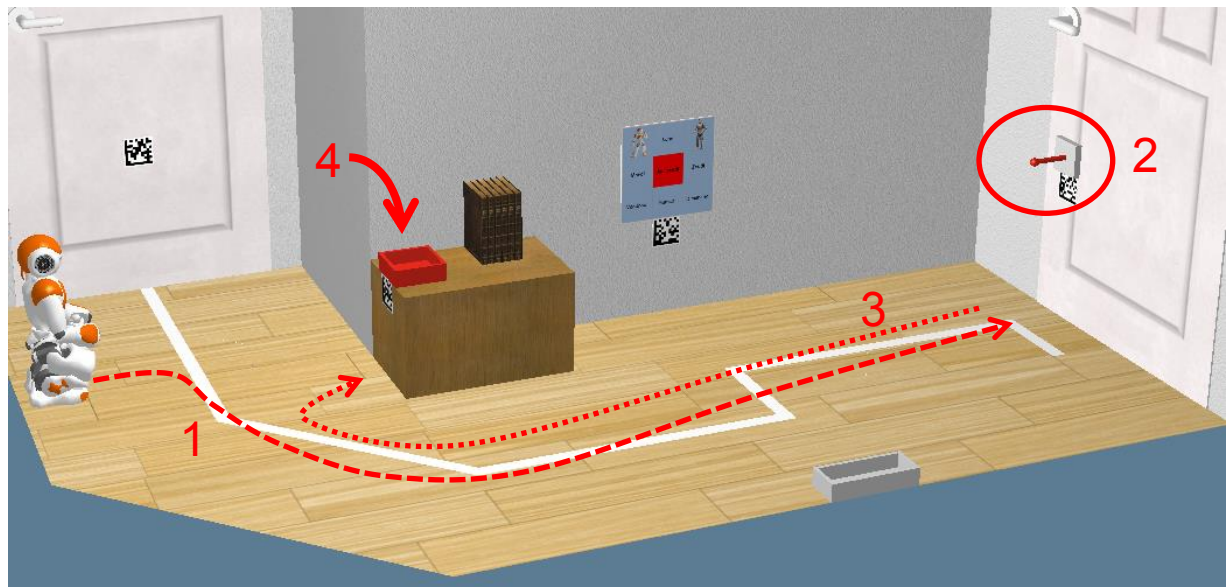
## DOCUMENTATION

- » Community
- » Forum

# NAO CHALLENGE

## LES EPREUVES

- » La clé
  - Aller chercher la clé sur la porte
  - La déposer dans le pot à clé sur la table



# NAO CHALLENGE

## LES EPREUVES

- » Le calendrier
  - Représente les 7 jours de la semaine
  - Un événement associé à chaque jour, choisi par les équipes
  - Le jour courant est en rouge
- » Le but :
  1. Aller au calendrier
  2. Lire le jour courant (= la position de la case rouge)
  3. Annoncer l'événement lié par une méthode "fun" !



# NAO CHALLENGE

## LES EPREUVES

- » Le distributeur
  - À développer et à construire !
  - Type d'activation au choix
  
- » Le but :
  - Aller au distributeur
  - L'activer
  - 50g de croquettes tombent dans la gamelle





# NAO CHALLENGE

## LES EPREUVES – LES POINTS CLES

- » Navigation et localisation
- » Création de mouvement
- » Traitement d'image
- » Communication avec le robot

# NAO CHALLENGE

## LES EPREUVES – LES POINTS CLES

- » Navigation et localisation
  - À la base des 3 épreuves !
  - Eviter la marche prédéfinie « aveugle »
  - Méthode au choix, par exemple :
    - Suivi de la ligne blanche
    - Marcher en direction des marques
    - Triangulation (marques ou points clés)
    - ...
- » Création de mouvement
- » Traitement d'image
- » Communication avec le robot

# NAO CHALLENGE

## LES EPREUVES – LES POINTS CLES

- » Navigation et localisation
- » Création de mouvement
  - Attention à l'équilibre et à la répétabilité
  - La réussite dépend du bon placement du robot !
  - Difficile à faire avec le simulateur, mais possible
  - Création : Chapitre suivant !
- » Traitement d'image
- » Communication avec le robot



# NAO CHALLENGE

## LES EPREUVES – LES POINTS CLES

- » Navigation et localisation
- » Création de mouvement
- » Traitement d'image
  - Utilisation d'OpenCV
  - Pas de retour visuel sur le robot directement
  - Développement dans tous les langages de programmation
  - Utilisation ? Chapitre suivant !
  - Attention aux différences d'éclairage !
- » Communication avec le robot

# NAO CHALLENGE

## LES EPREUVES – LES POINTS CLES

- » Navigation et localisation
- » Création de mouvement
- » Traitement d'image
- » Communication avec le robot
  - Robot autonome ?
    - Programme en Choregraphe, Python ou C++ uniquement
    - Difficile à déboguer !
    - Plus « réel »
  - Robot commandé à distance par l'ordinateur (wifi)
    - Aucune action autorisée sur l'ordinateur pendant l'épreuve
    - Programmation dans tous les langages de programmation
    - Attention à la qualité du signal du wifi !



# NAO CHALLENGE

## LES EPREUVES – LES POINTS CLES

- » Navigation et localisation
- » Création de mouvement
- » Traitement d'image
- » Communication avec le robot

Le règlement, les épreuves, comment ? Pourquoi ? ... ?

## AVEZ-VOUS DES QUESTIONS ?

# DEMANDEZ LE PROGRAMME !

## PREMIERS PAS AVEC NAO

- » Le robot
- » Choregraphe, Webots
- » Le SDK

## NAO CHALLENGE

- » Retour sur le règlement
- » Détails des épreuves

## EXEMPLES APPLIQUES

- » OpenCV
- » Création de mouvements

## CONCLUSION

- » Questions, Réponses !



# DEMANDEZ LE PROGRAMME !

## PREMIERS PAS AVEC NAO

- » Le robot
- » Choregraphe, Webots
- » Le SDK

## NAO CHALLENGE

- » Retour sur le règlement
- » Détails des épreuves

## EXEMPLES APPLIQUES

- » OpenCV
- » Création de mouvements

## CONCLUSION

- » Questions, Réponses !



# PROGRAMMONS !

## ELEMENTS NECESSAIRES POUR LE CHALLENGE

1. Traitement d'image
  - OpenCV, sur l'ordinateur et sur le robot
  - Lien avec Chorégraphe
  - Suivi de la ligne, lecture du calendrier, recherche de la clé, dépôt précis dans le pot, ...
  
2. Création d'un mouvement
  - Simple avec NAO, mais avec Webots ?
  - Ajustement précis d'un mouvement
  - Prendre la clé, la déposer, activer le distributeur...

## RÉCUPÉRER L'IMAGE DES CAMÉRAS

```
from naoqi import *
from vision_definitions import *
```

```
naoIP      = "supernao.local"
moduleName = "OpenCVDemo"
camera     = 1    # 1 = Bas/ 0 = Haut
```

Nom utilisé par NAOqi par la suite

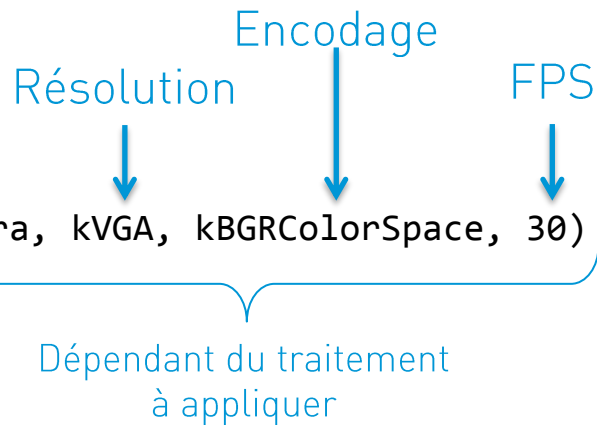
```
# creer un proxy vers le module ALVideoDevice
videoProxy = ALProxy("ALVideoDevice", naoIP, 9559)
```

```
# specifier au module le type d'image requis
```

```
moduleName = videoProxy.subscribeCamera(moduleName, camera, kVGA, kBGRColorSpace, 30)
```

```
# obtenir une image
```

```
ImageNAO = videoProxy.getImageRemote(moduleName)
```



## CONVERSION DE L'IMAGE

```
from cv2 import *  
from numpy import *
```

← À ajouter au début

[...]

# obtenir une image

```
ImageNAO = videoProxy.getImageRemote(moduleName)
```

# lire les parametres

```
largeur = ImageNAO[0]  
hauteur = ImageNAO[1]  
canaux = ImageNAO[2]  
imageBrute = ImageNAO[6]
```

# Conversion « magique » vers OpenCV

```
imageBGR = fromstring(imageBrute, dtype="uint8").reshape(hauteur, largeur, canaux)
```

*À apprendre par cœur !*



[...]

```
# affichage de l'image originale
```

```
imshow("Camera", imageBGR)
```

```
# exemple de traitement : flou
```

```
imageFlou = blur(imageBGR, (10,10))
```

```
# affichage du resultat
```

```
imshow("Camera", imageFlou)
```

» Pour le voir en « live » : intégrer dans une boucle

```
from naoqi import *
from vision_definitions import *
from cv2 import *
from numpy import *

naoIP      = "supernao.local"
moduleName = "OpenCVDemo"
camera     = 1    # 1 = Bas/ 0 = Haut

# creer un proxy vers le module ALVideoDevice
videoProxy = ALProxy("ALVideoDevice", naoIP, 9559)

# specifier au module le type d'image requis
moduleName = videoProxy.subscribeCamera(moduleName,
camera, kVGA, kBGRColorSpace, 30)

key = 0
```

```
while(key != 27):
    # obtenir une image
    ImageNAO = videoProxy.getImageRemote(moduleName)

    # lire les parametres
    largeur  = ImageNAO[0]
    hauteur  = ImageNAO[1]
    canaux    = ImageNAO[2]
    imageBrute = ImageNAO[6]

    # Conversion « magique » vers OpenCV
    imageBGR = fromstring(imageBrute,
dtype="uint8").reshape(hauteur, largeur, canaux)

    # affichage de l'image
    imshow("Camera", imageBGR)

    # Appui sur ECHAP ?
    key = waitKey(1)

# Cleanup
destroyAllWindows()
videoProxy.unsubscribe(moduleName)
```

```
from naoqi import *
from vision_definitions import *
from cv2 import *
from numpy import *

naoIP      = "supernao.local"
moduleName = "OpenCVDemo"
camera    = 1  # 1 = Bas/ 0 = Haut

# creer un proxy vers le module ALVideoDevice
videoProxy = ALProxy("ALVideoDevice", naoIP, 9559)

# specifier au module le type d'image requis
moduleName = videoProxy.subscribeCamera(moduleName,
camera, kVGA, kBGRCOLORSPACE, 30)

key = 0
```

Insérez votre traitement ici →

```
while(key != 27):
    # obtenir une image
    ImageNAO = videoProxy.getImageRemote(moduleName)

    # lire les parametres
    largeur  = ImageNAO[0]
    hauteur  = ImageNAO[1]
    canaux    = ImageNAO[2]
    imageBrute = ImageNAO[6]

    # Conversion « magique » vers OpenCV
    imageBGR = fromstring(imageBrute,
dtype="uint8").reshape(hauteur, largeur, canaux)

    # affichage de l'image
    imshow("Camera", imageBGR)

    # Appui sur ECHAP ?
    key = waitKey(1)

# Cleanup
destroyAllWindows()
videoProxy.unsubscribe(moduleName)
```

```
from naoqi import *
from vision_definitions import *
from cv2 import *
from numpy import *

naoIP      = "supernao.local"
moduleName = "OpenCVDemo"
camera     = 1  # 1 = Bas/ 0 = Haut

# creer un proxy vers le module ALVideoDevice
videoProxy = ALProxy("ALVideoDevice", naoIP, 9559)

# specifier au module le type d'image requis
moduleName = videoProxy.subscribeCamera(moduleName,
camera, kVGA, kBGRColorSpace, 30)

key = 0
```

Insérez votre traitement ici →

```
while(key != 27):
    # obtenir une image
    ImageNAO = videoProxy.getImageRemote(moduleName)

    # lire les parametres
    largeur  = ImageNAO[0]
    hauteur  = ImageNAO[1]
    canaux    = ImageNAO[2]
    imageBrute = ImageNAO[6]

    # Conversion « magique » vers OpenCV
    imageBGR = fromstring(imageBrute,
dtype="uint8").reshape(hauteur, largeur, canaux)

    # affichage de l'image
    imshow("Camera", imageBGR)

    # Appui sur ECHAP ?
    key = waitKey(1)

# Cleanup
destroyAllWindows()
videoProxy.unsubscribe(moduleName)
```

# OPEN CV

## LIEN AVEC CHOREGRAPHE

- » Transmettre les résultats via ALMemory
  - ALMemory → insertData( nom , valeur )
  - ALMemory → getData( nom )
  
- » Intégrer l'algorithme de traitement dans une boite
  - ALProxy( module , ~~robot~~ , ~~port~~ )
  - Envoyer les résultats vers les sorties de la boite
  - Supprimer les affichages et lectures de clavier !
  - Fonctionnement identique !



# OPEN CV

AVEZ-VOUS DES QUESTIONS ?

# CREATION DE MOUVEMENT

- » Mouvement = suite de positions à des temps données. NAO fait l'interpolation entre les positions.

# CREATION DE MOUVEMENT

- » Mouvement = suite de positions à des temps données. NAO fait l'interpolation entre les positions.

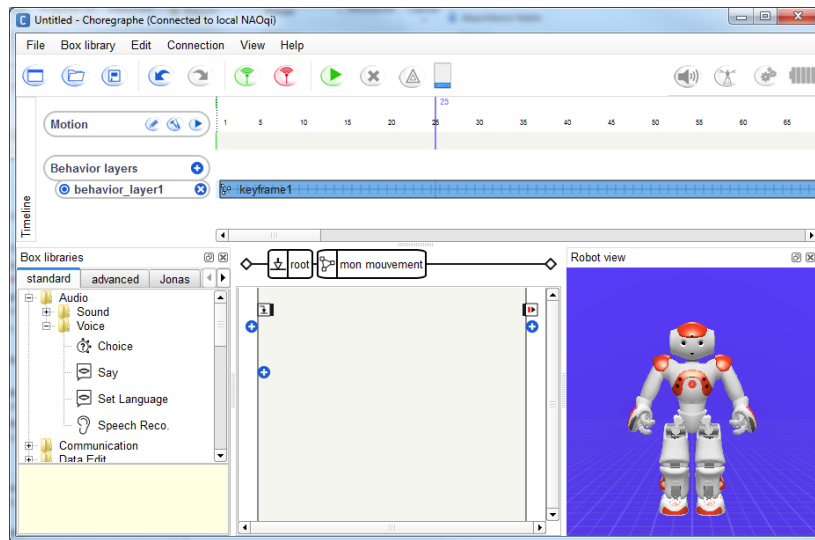
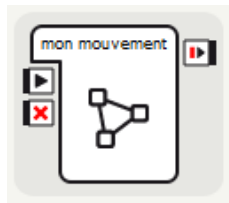
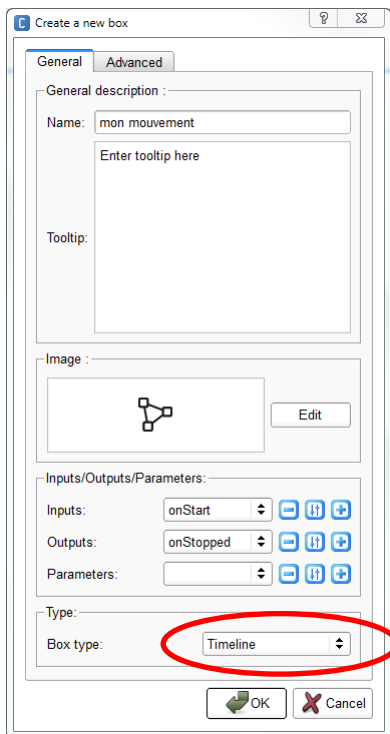
## PLUSIEURS METHODES :

- » **Script** : coder les positions de chaque moteur au cours du temps
- » **Timeline** : enregistrer les positions
  - En bougeant NAO manuellement
  - Avec le mode « Animation »
  - En faisant bouger le NAO simulé



# CREATION DE MOUVEMENT

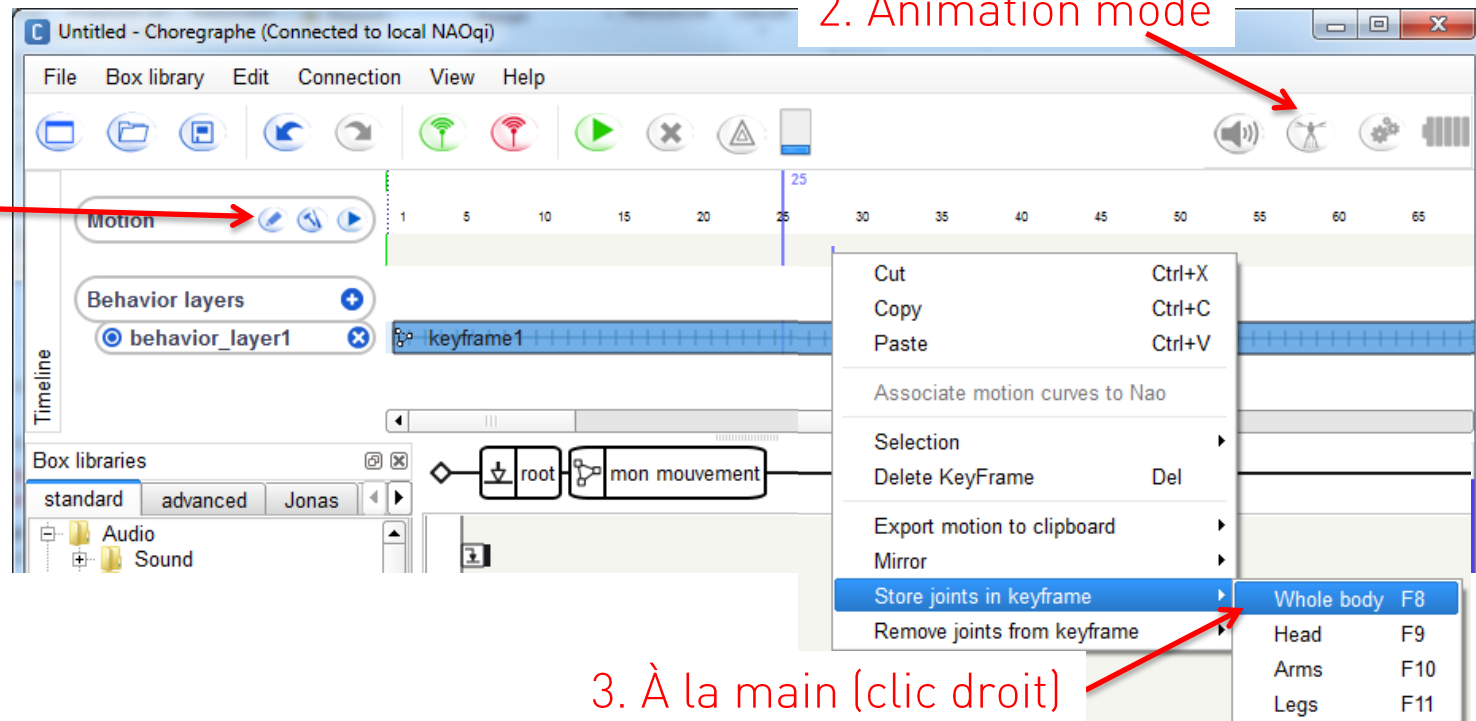
## CRÉER UNE NOUVELLE BOITE



# CREATION DE MOUVEMENT

## ENREGISTRER UNE POSITION...

1. Timeline editor

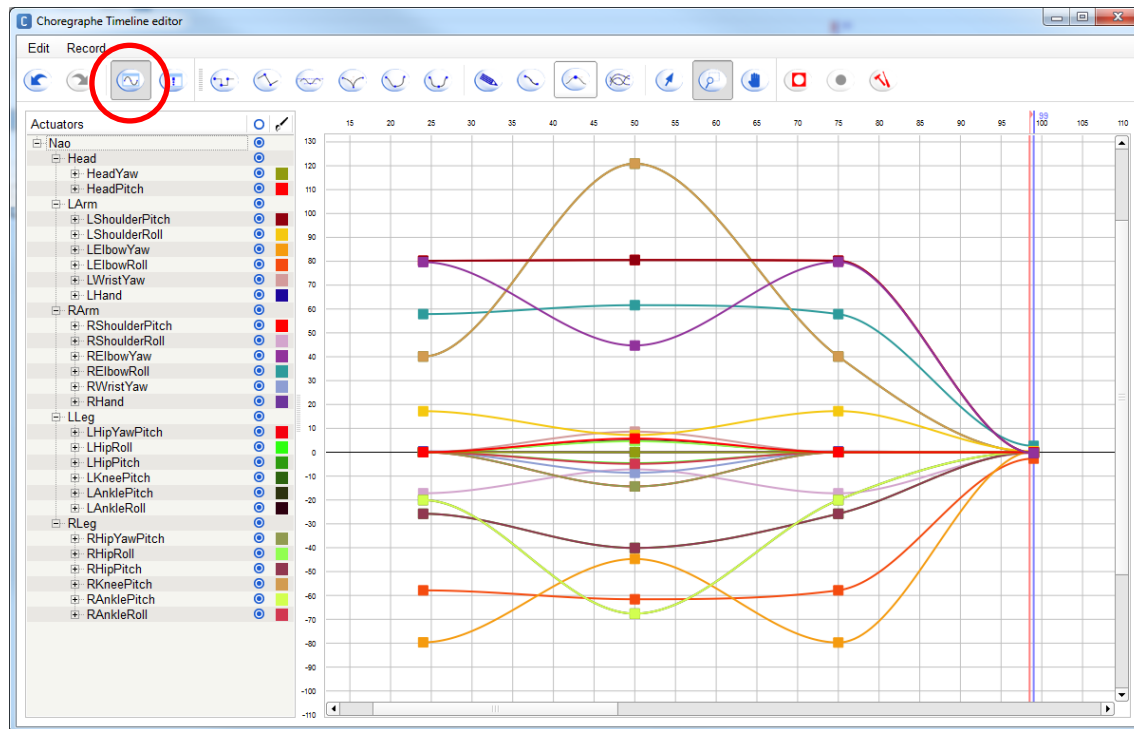


2. Animation mode

3. À la main (clic droit)

# CREATION DE MOUVEMENT

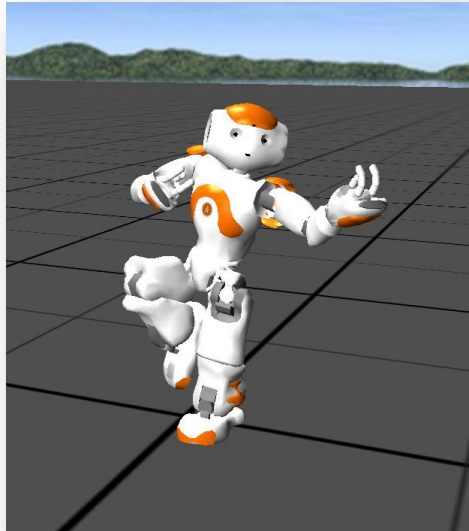
## EDITER UN MOUVEMENT



# CREATION DE MOUVEMENT

À VOTRE TOUR !

- » Créez un mouvement **stable**
- » Mouvement sur un pied !





# CREATION DE MOUVEMENT

AVEZ-VOUS DES QUESTIONS ?

# DEMANDEZ LE PROGRAMME !

## PREMIERS PAS AVEC NAO

- » Le robot
- » Choregraphe, Webots
- » Le SDK

## NAO CHALLENGE

- » Retour sur le règlement
- » Détails des épreuves

## EXEMPLES APPLIQUES

- » OpenCV
- » Création de mouvements

## CONCLUSION

- » Questions, Réponses !



# DEMANDEZ LE PROGRAMME !

## PREMIERS PAS AVEC NAO

- » Le robot
- » Choregraphe, Webots
- » Le SDK

## NAO CHALLENGE

- » Retour sur le règlement
- » Détails des épreuves

## EXEMPLES APPLIQUES

- » OpenCV
- » Création de mouvements

## CONCLUSION

- » Questions, Réponses !





# CONCLUSION

## PREMIERS PAS AVEC NAO

- » Le robot
- » Choregraphe, Webots
- » Le SDK

## NAO CHALLENGE

- » Retour sur le règlement
- » Détails des épreuves

## EXEMPLES APPLIQUES

- » OpenCV
- » Création de mouvements

## CONCLUSION

- » Questions, Réponses !

AVEZ-VOUS  
DES  
QUESTIONS ?



# À VOTRE TOUR !

- » Rendez-vous sur les forums et aux compétitions !
- » N'hésitez pas à poser vos questions !

## BON COURAGE !

