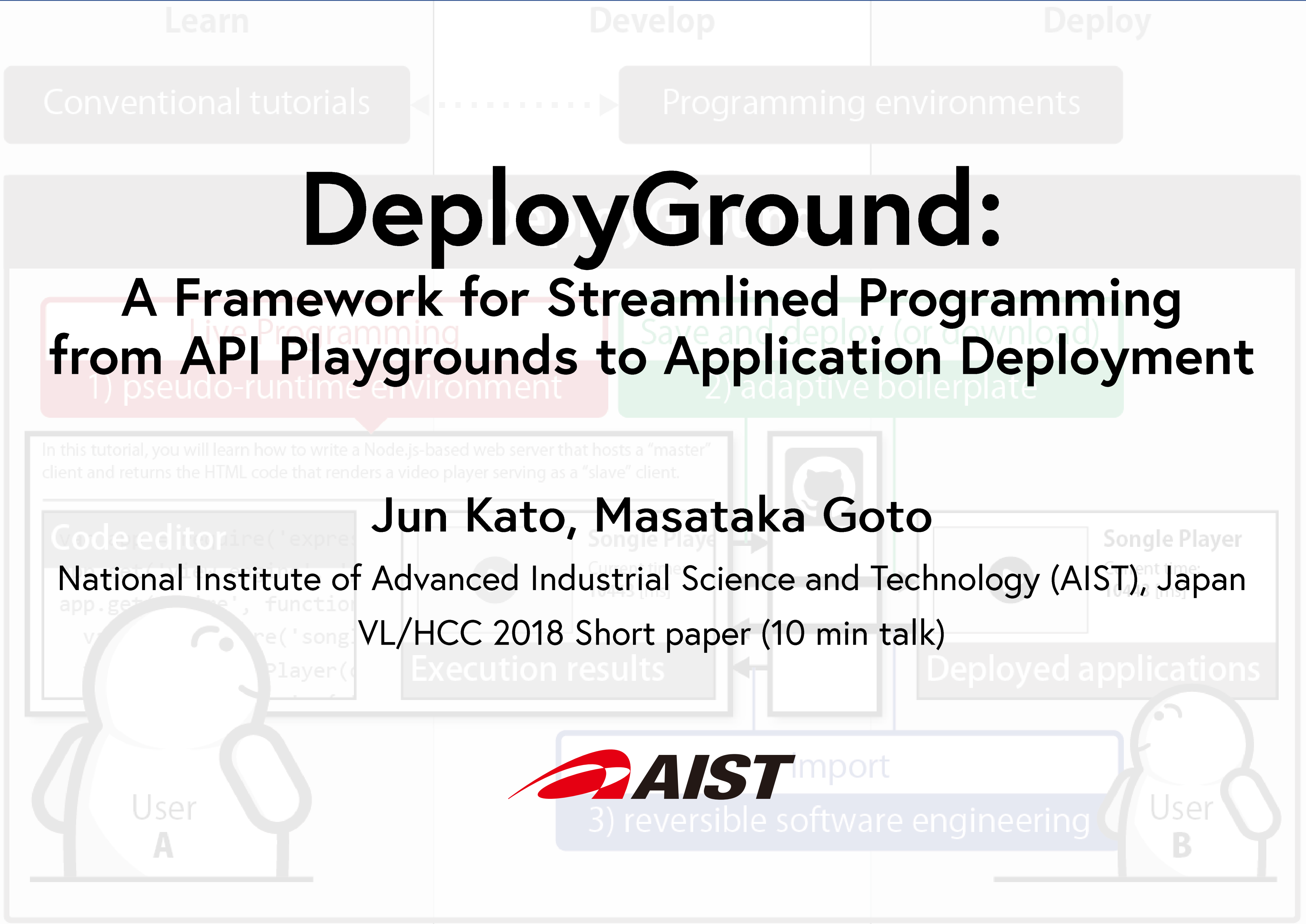

Learn
Develop
Deploy
Conventional tutorials
Programming environments
Live Programming
1) pseudo-runtime environment
Save and deploy (or download)
2) adaptive boilerplate
In this tutorial, you will learn how to write a Node.js-based web server that hosts a "master" client and returns the HTML code that renders a video player serving as a "slave" client.
Code editor
Execution results
Songle Player
Deployed applications
Import
3) reversible software engineering
User
A
User
B

DeployGround:

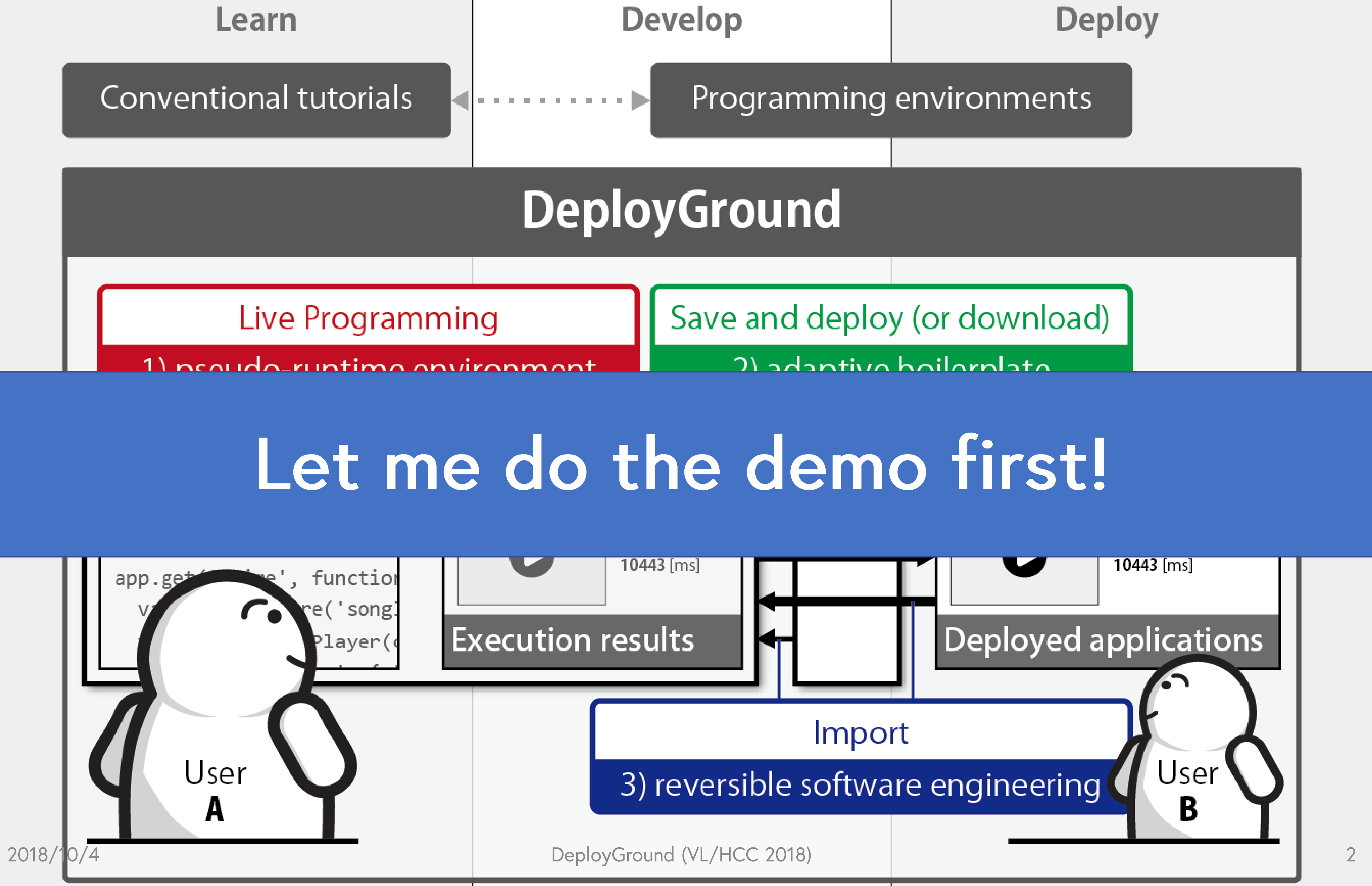
A Framework for Streamlined Programming from API Playgrounds to Application Deployment

Jun Kato, Masataka Goto

National Institute of Advanced Industrial Science and Technology (AIST), Japan

VL/HCC 2018 Short paper (10 min talk)

AIST



Learn

Develop

Deploy

Conventional tutorials

Programming environments

DeployGround

Live Programming

1) pseudo-runtime environment

Save and deploy (or download)

2) adaptive boilerplate

Execution results

Deployed applications

Import

3) reversible software engineering

Let me do the demo first!

BACKGROUND:

Coding tutorials and references

- Much work on creating tutorials in the context of HCI
  [Chi et al., UIST '12] [Lafreniere et al., CHI '13] [Chi et al., UIST '13] [Kim et al., CHI '14]
- Only a handful of work on creating coding tutorials
  [Harms et al., IDC '13] [Head et al., VL/HCC '15] [Gordon et al., VL/HCC '15]

See https://junkato.jp/deployground for the complete list of references.

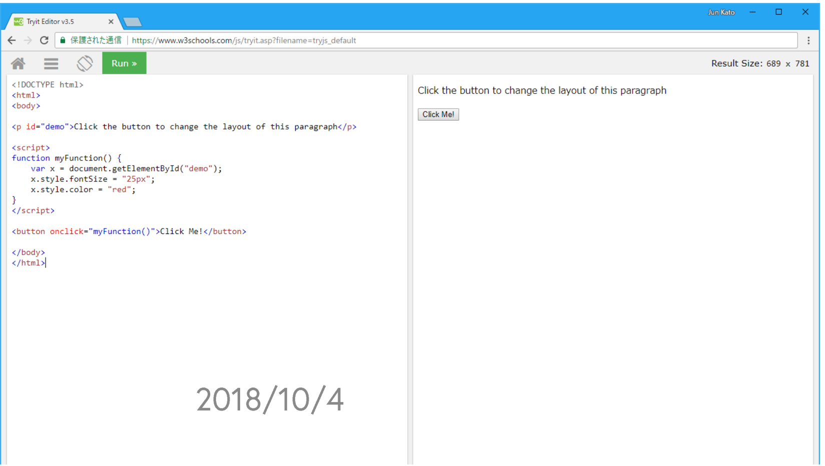
Contributions:

- A framework to create interactive web-based coding tutorials
- Its concrete implementation techniques

BACKGROUND:

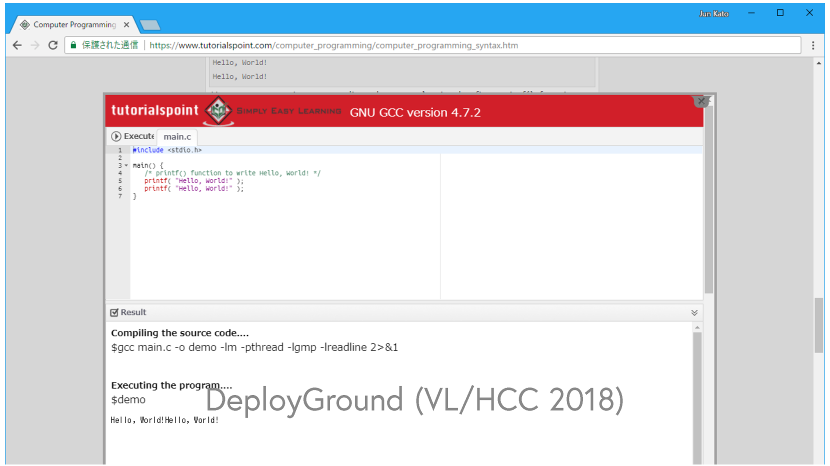
API documentations and tutorials

- Concrete usage information is beneficial
  [Robillard, 2009] [Hou et al., ICPC '11] [Robillard et al., 2011] [Wang et al., MSR '13]
- Executable example codes are especially beneficial
  [Subramanian et al., ICSE '14]
- Some documentations and tutorials provide "playgrounds"
  [Khan Academy] [TypeScript Playground] [Vimeo API Playground] [W3School] [tutorialspoint] …


Tryit Editor v3.5
Jun Kato
保護された通信 | https://www.w3schools.com/js/tryit.asp?filename=tryjs_default
Run »
Result Size: 689 x 781

```
<!DOCTYPE html>
<html>
<body>

<p id="demo">Click the button to change the layout of this paragraph</p>

<script>
function myFunction() {
    var x = document.getElementById("demo");
    x.style.fontSize = "25px";
    x.style.color = "red";
}
</script>

<button onclick="myFunction()">Click Me!</button>

</body>
</html>
```

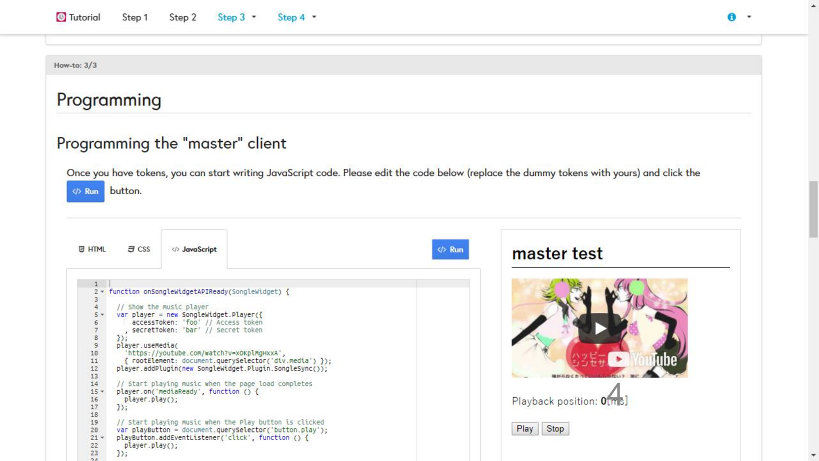
Click the button to change the layout of this paragraph


Computer Programming
Jun Kato
保護された通信 | https://www.tutorialspoint.com/computer_programming/computer_programming_syntax.htm

```
Hello, World!
Hello, World!
```

```
#include <stdio.h>

main() {
   /* printf() function to write Hello, World! */
   printf( "Hello, World!" );
   printf( "Hello, World!" );
}
```

```
Compiling the source code....
$gcc main.c -o demo -lm -pthread -lgmp -lreadline 2>&1


Executing the program....
$demo

Hello, World!Hello, World!
```



Programming

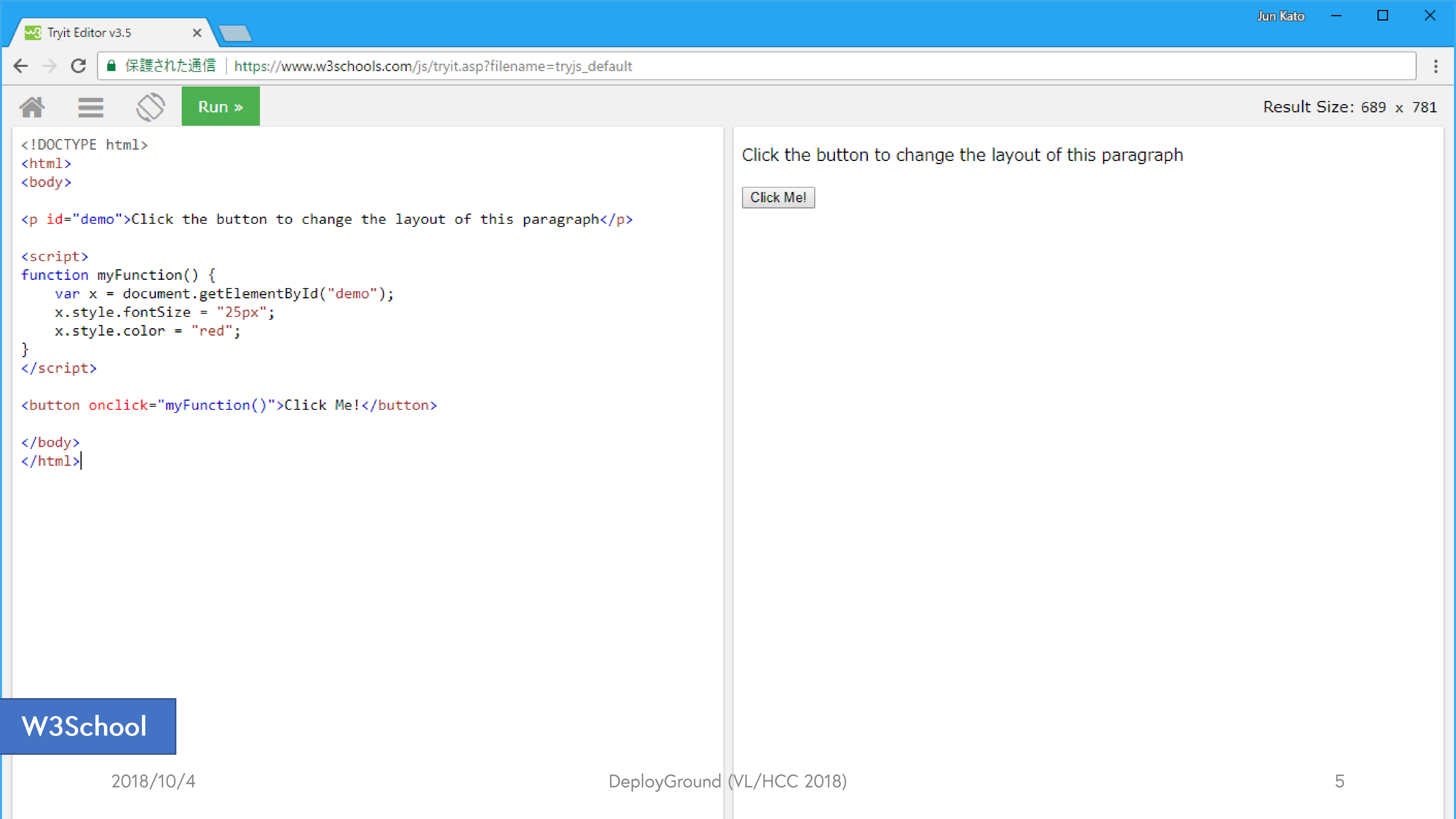
Programming the "master" client

Once you have tokens, you can start writing JavaScript code. Please edit the code below (replace the dummy tokens with yours) and click the Run button.

```

function onSonglewidgetAPIReady(SongleWidget) {

  // Show the music player
  var player = new SongleWidget.Player({
      accessToken: 'foo' // Access token
    , secretToken: 'bar' // Secret token
  });
  player.useMedia(
    'https://youtube.com/watch?v=xOKplMgHxxA',
    { rootElement: document.querySelector('div.media') });
  player.addPlugin(new SongleWidget.Plugin.SongleSync());

  // Start playing music when the page load completes
  player.on('mediaReady', function () {
    player.play();
  });

  // Start playing music when the Play button is clicked
  var playButton = document.querySelector('button.play');
  playButton.addEventListener('click', function () {
    player.play();
  });
```

master test

Playback position: 0[ms]



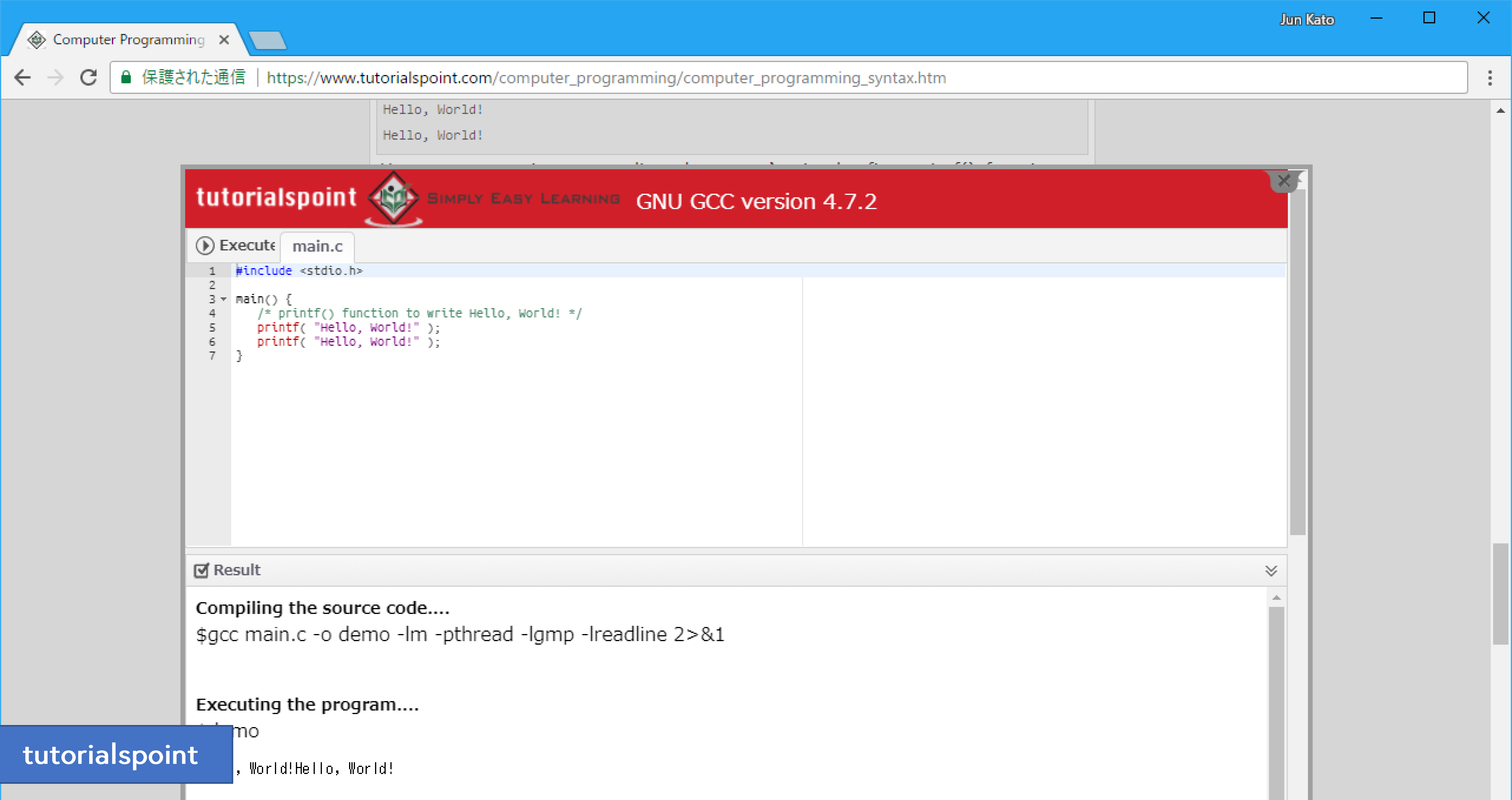
https://www.w3schools.com/js/tryit.asp?filename=tryjs_default

Result Size: 689 x 781

```
<!DOCTYPE html>
<html>
<body>

<p id="demo">Click the button to change the layout of this paragraph</p>

<script>
function myFunction() {
    var x = document.getElementById("demo");
    x.style.fontSize = "25px";
    x.style.color = "red";
}
</script>

<button onclick="myFunction()">Click Me!</button>

</body>
</html>
```

Click the button to change the layout of this paragraph

W3School



```
Hello, World!
Hello, World!
```

GNU GCC version 4.7.2

```c
#include <stdio.h>

main() {
   /* printf() function to write Hello, World! */
   printf( "Hello, World!" );
   printf( "Hello, World!" );
}
```

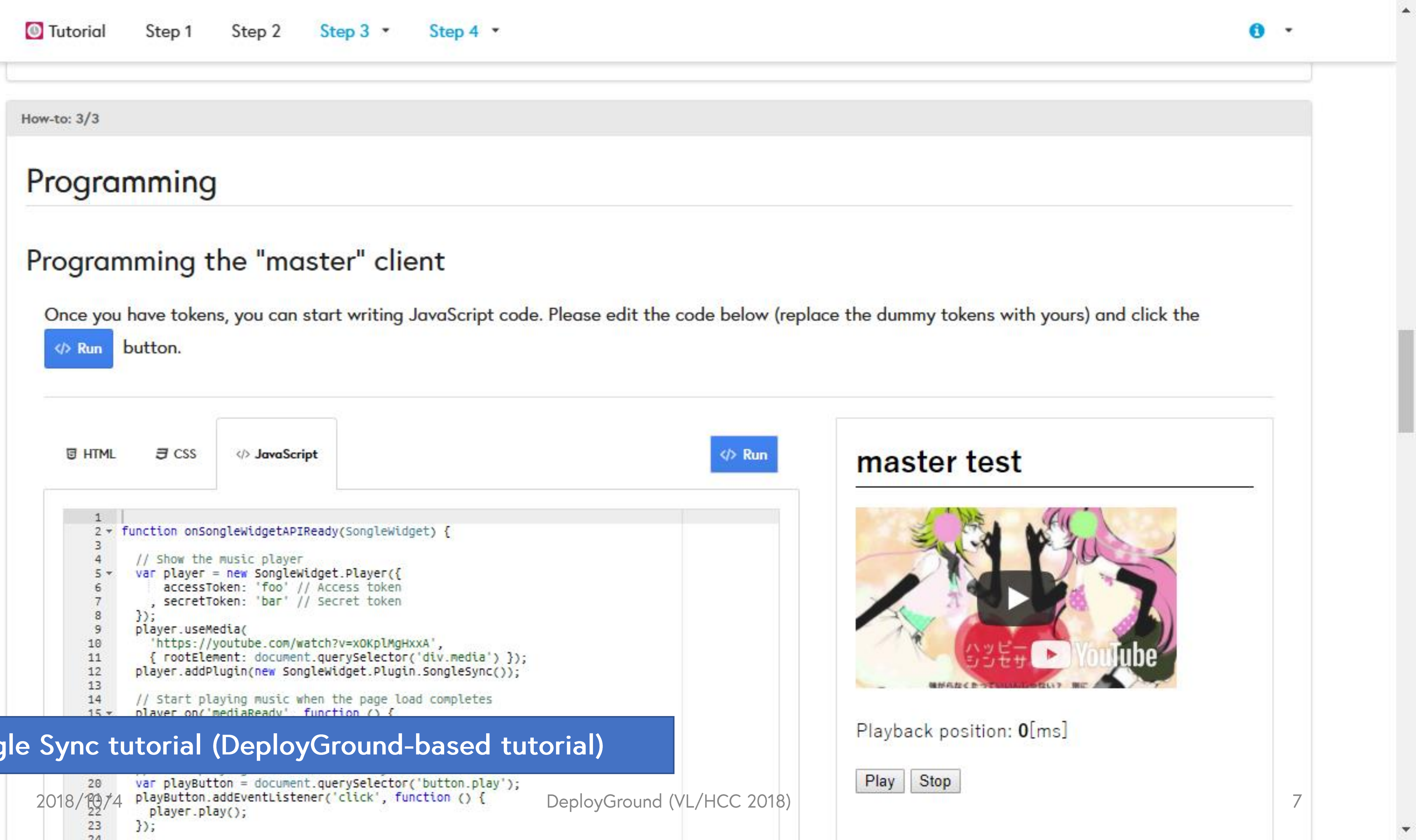
Compiling the source code....

```
$gcc main.c -o demo -lm -pthread -lgmp -lreadline 2>&1
```

Executing the program....

```
mo
, World!Hello, World!
```

tutorialspoint



Programming

Programming the "master" client

Once you have tokens, you can start writing JavaScript code. Please edit the code below (replace the dummy tokens with yours) and click the </> Run button.

```

function onSongleWidgetAPIReady(SongleWidget) {

  // Show the music player
  var player = new SongleWidget.Player({
      accessToken: 'foo' // Access token
    , secretToken: 'bar' // Secret token
  });
  player.useMedia(
    'https://youtube.com/watch?v=xOKplMgHxxA',
    { rootElement: document.querySelector('div.media') });
  player.addPlugin(new SongleWidget.Plugin.SongleSync());

  // Start playing music when the page load completes
  player.on('mediaReady', function () {
```

```
  var playButton = document.querySelector('button.play');
  playButton.addEventListener('click', function () {
    player.play();
  });
```

master test

Playback position: **0**[ms]

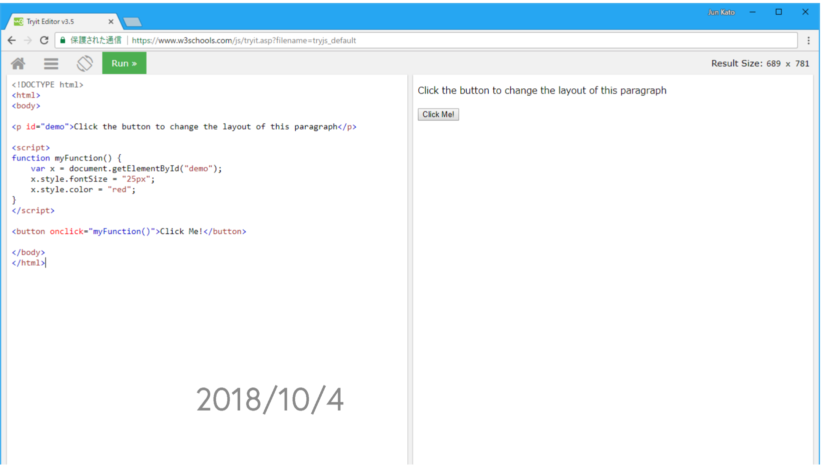
Play Stop

Songle Sync tutorial (DeployGround-based tutorial)

BACKGROUND:

API Playgrounds

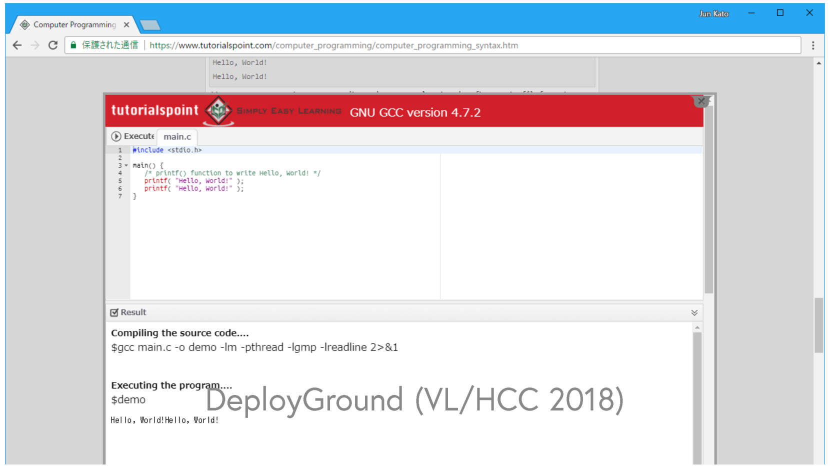
- A part of coding tutorials (API documentations or tutorials)
- A code editor and its output sit next to each other
- The output can be interactively updated upon the user's request



Tryit Editor v3.5
Jun Kato
保護された通信 | https://www.w3schools.com/js/tryit.asp?filename=tryjs_default
Run »
Result Size: 689 x 781

```
<!DOCTYPE html>
<html>
<body>

<p id="demo">Click the button to change the layout of this paragraph</p>

<script>
function myFunction() {
    var x = document.getElementById("demo");
    x.style.fontSize = "25px";
    x.style.color = "red";
}
</script>

<button onclick="myFunction()">Click Me!</button>

</body>
</html>
```

Click the button to change the layout of this paragraph

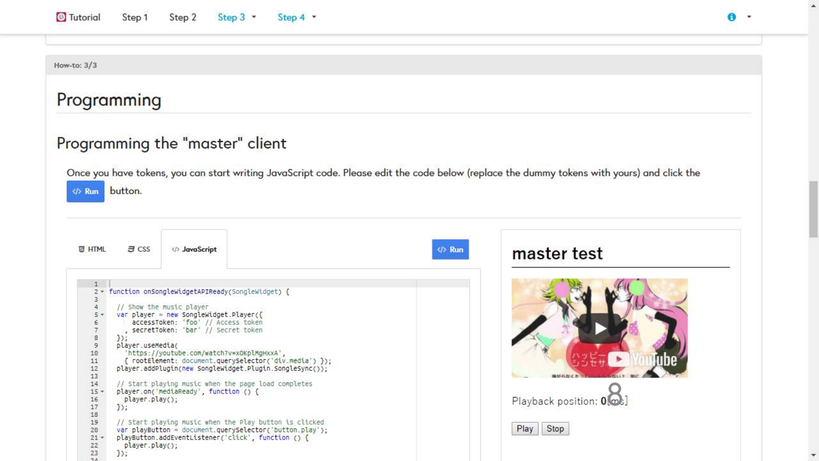


Computer Programming
Jun Kato
保護された通信 | https://www.tutorialspoint.com/computer_programming/computer_programming_syntax.htm

```
Hello, World!
Hello, World!
```

```
#include <stdio.h>

main() {
   /* printf() function to write Hello, World! */
   printf( "Hello, World!" );
   printf( "Hello, World!" );
}
```

```
Compiling the source code....
$gcc main.c -o demo -lm -pthread -lgmp -lreadline 2>&1


Executing the program....
$demo

Hello, World!Hello, World!
```



Programming

Programming the "master" client

Once you have tokens, you can start writing JavaScript code. Please edit the code below (replace the dummy tokens with yours) and click the Run button.

```

function onSongleWidgetAPIReady(SongleWidget) {

  // Show the music player
  var player = new SongleWidget.Player({
      accessToken: 'foo' // Access token
    , secretToken: 'bar' // Secret token
  });
  player.useMedia(
    'https://youtube.com/watch?v=xOKplMgHxxA',
    { rootElement: document.querySelector('div.media') });
  player.addPlugin(new SongleWidget.Plugin.SongleSync());

  // Start playing music when the page load completes
  player.on('mediaReady', function () {
    player.play();
  });

  // Start playing music when the Play button is clicked
  var playButton = document.querySelector('button.play');
  playButton.addEventListener('click', function () {
    player.play();
  });
```

master test

Playback position: 0[ms]

MOTIVATION:

Limitations of existing tutorials

- Learning APIs is supported well with the interactive playgrounds

meanwhile…

- The programmer needs to leave the tutorial at some point
- S/he needs to re-start the development in their own environment.

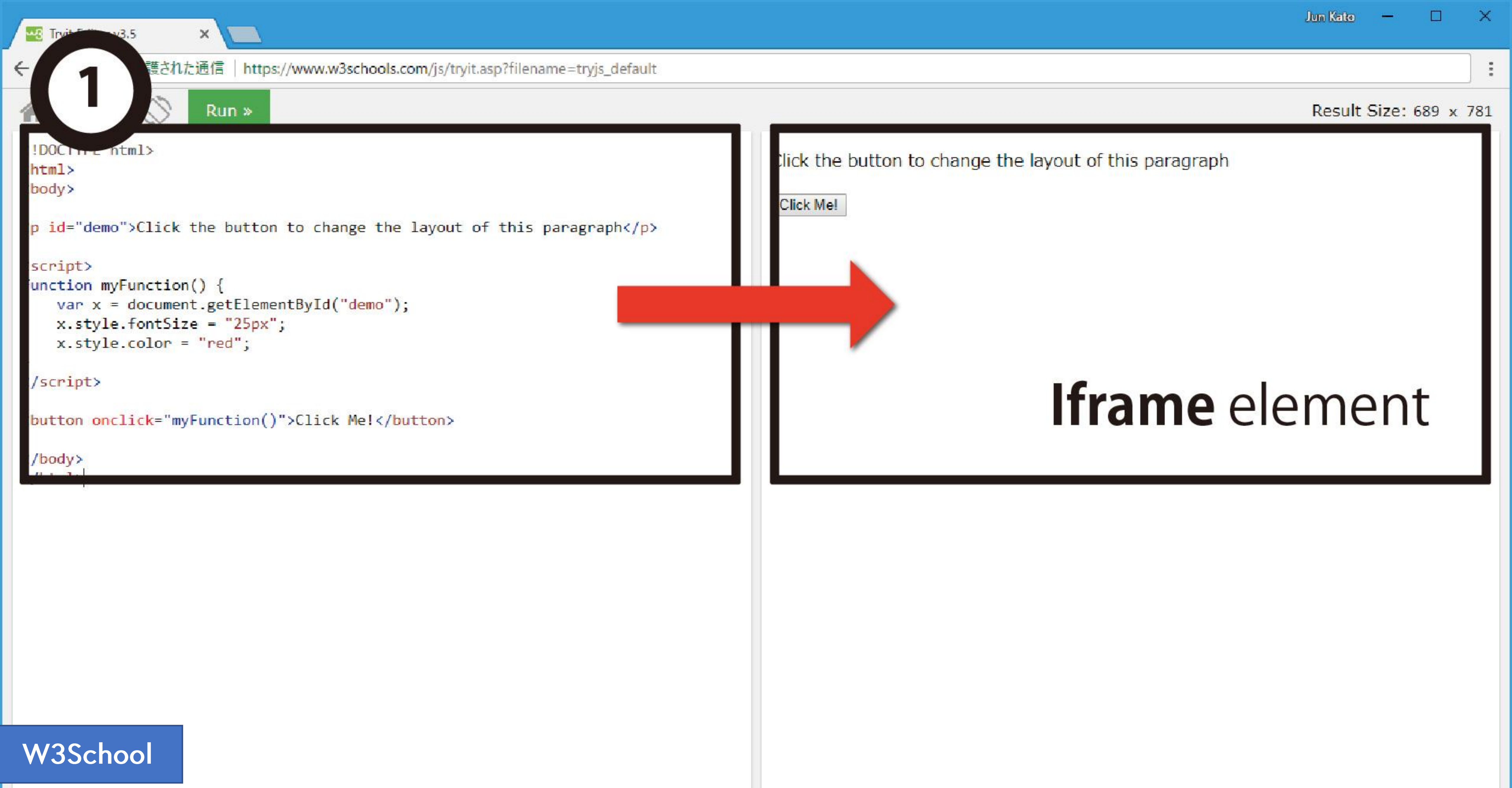
Support for seamless post-learning experience is in need

MOTIVATION:

Detailed limitations of existing tutorials

- Toy sandbox OR expensive sandbox
- Ephemeral code
- No support for deployment
- Little social interaction

What are these and how can we address them?

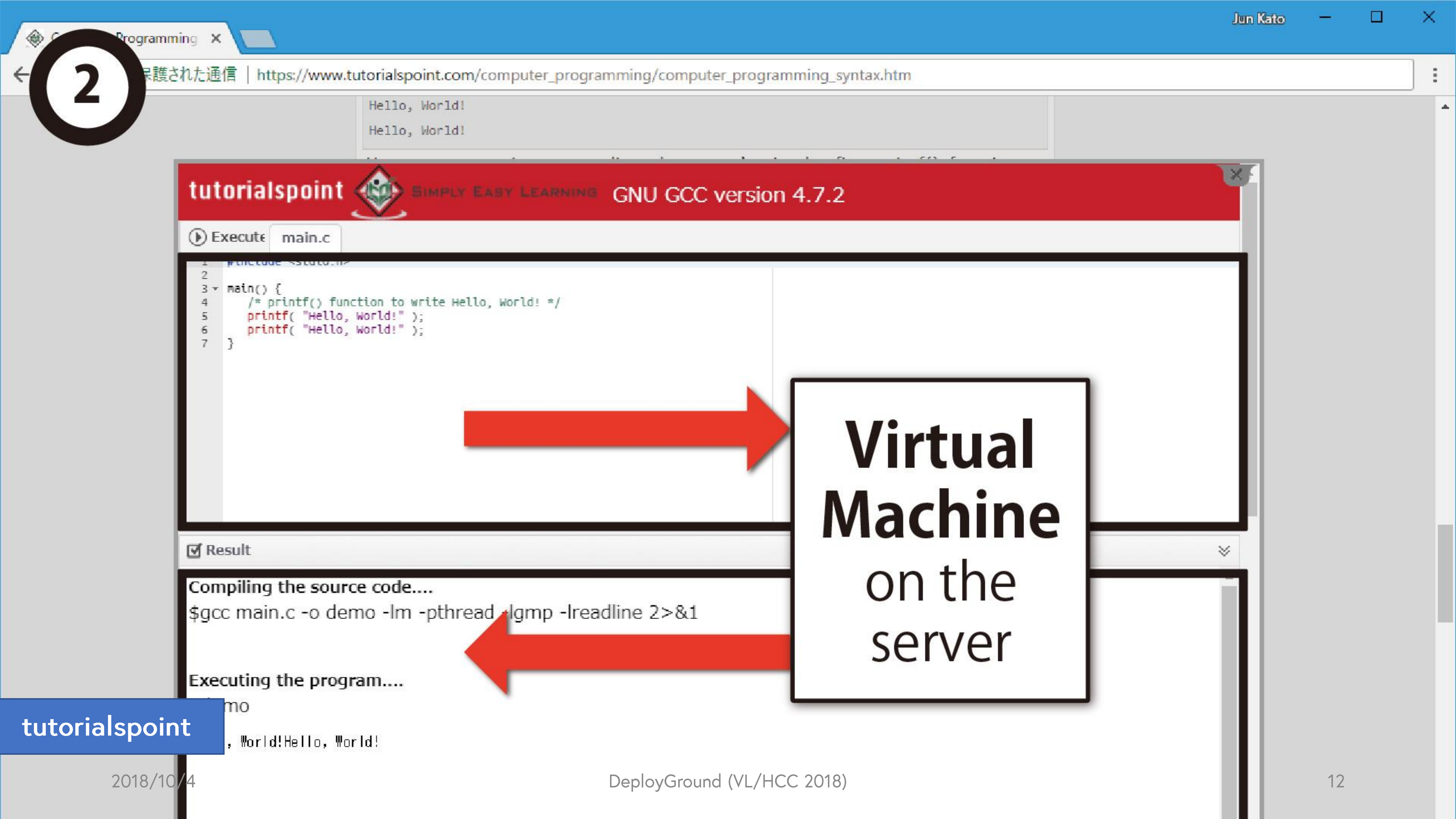


```
!DOCTYPE html>
html>
body>

p id="demo">Click the button to change the layout of this paragraph</p>

script>
unction myFunction() {
   var x = document.getElementById("demo");
   x.style.fontSize = "25px";
   x.style.color = "red";

/script>

button onclick="myFunction()">Click Me!</button>

/body>
/html>
```

lick the button to change the layout of this paragraph

Click Me!

W3School



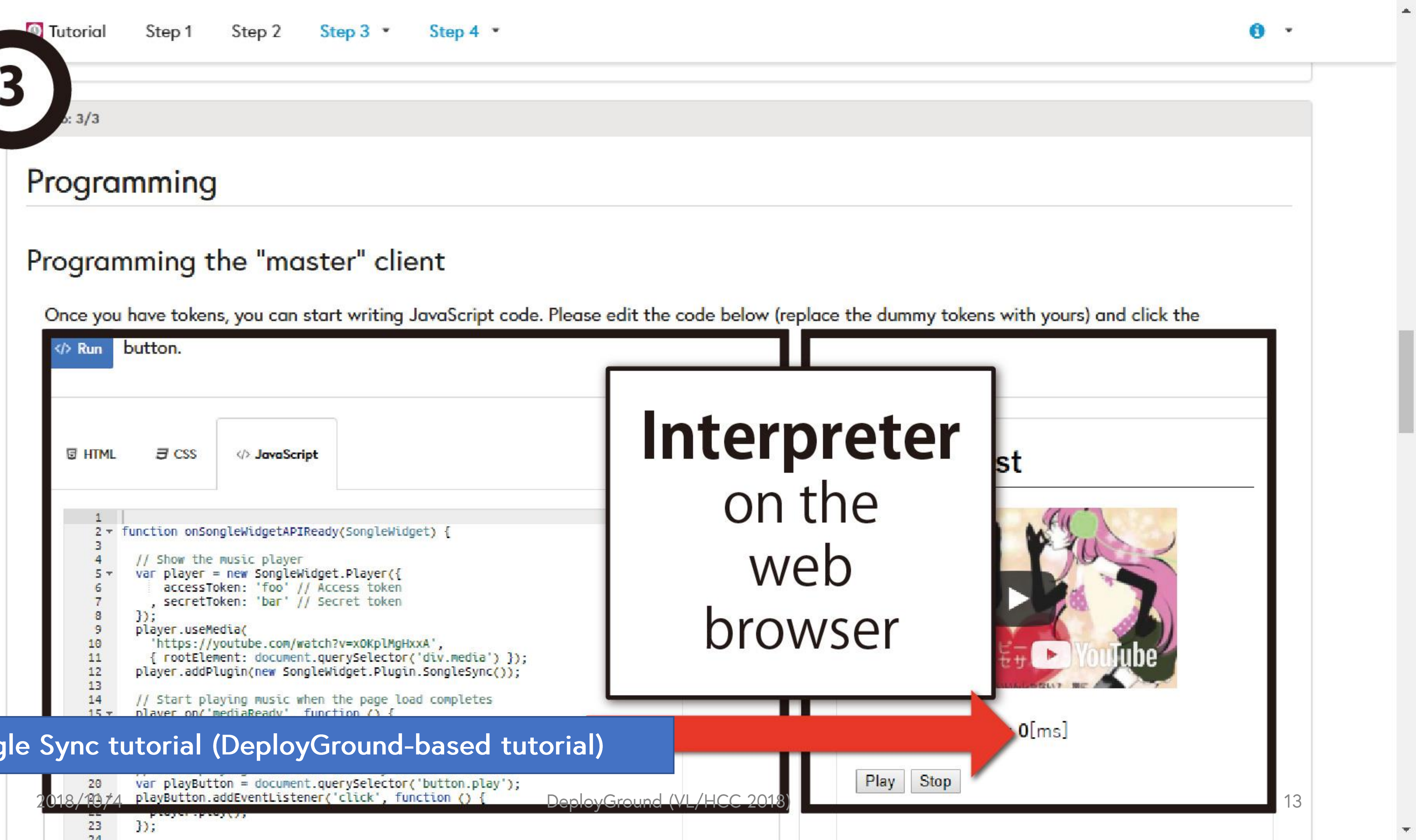
```
Hello, World!
Hello, World!
```

```
#include <stdio.h>

main() {
   /* printf() function to write Hello, World! */
   printf( "Hello, World!" );
   printf( "Hello, World!" );
}
```

```
Compiling the source code....
$gcc main.c -o demo -lm -pthread -lgmp -lreadline 2>&1


Executing the program....
  mo

, World!Hello, World!
```

Virtual Machine on the server

3



Programming

Programming the "master" client

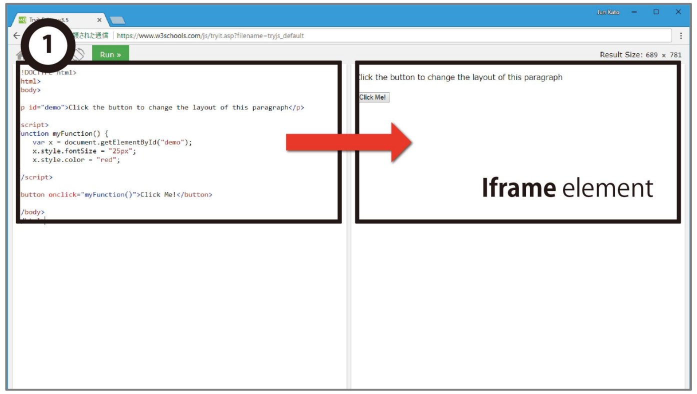
Once you have tokens, you can start writing JavaScript code. Please edit the code below (replace the dummy tokens with yours) and click the </> Run button.

```
function onSongleWidgetAPIReady(SongleWidget) {

  // Show the music player
  var player = new SongleWidget.Player({
      accessToken: 'foo' // Access token
    , secretToken: 'bar' // Secret token
  });
  player.useMedia(
    'https://youtube.com/watch?v=xOKplMgHxxA',
    { rootElement: document.querySelector('div.media') });
  player.addPlugin(new SongleWidget.Plugin.SongleSync());

  // Start playing music when the page load completes
  player.on('mediaReady', function () {

  var playButton = document.querySelector('button.play');
  playButton.addEventListener('click', function () {
    player.play();
  });
```

Interpreter on the web browser

Songle Sync tutorial (DeployGround-based tutorial)

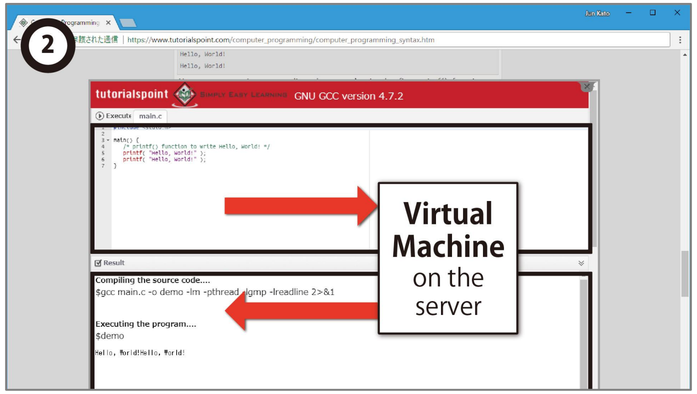
Toy sandbox OR expensive sandbox?



https://www.w3schools.com/js/tryit.asp?filename=tryjs_default

```
!DOCTYPE html>
html>
body>

p id="demo">Click the button to change the layout of this paragraph</p>

script>
unction myFunction() {
   var x = document.getElementById("demo");
   x.style.fontSize = "25px";
   x.style.color = "red";

/script>

button onclick="myFunction()">Click Me!</button>

/body>
```

lick the button to change the layout of this paragraph

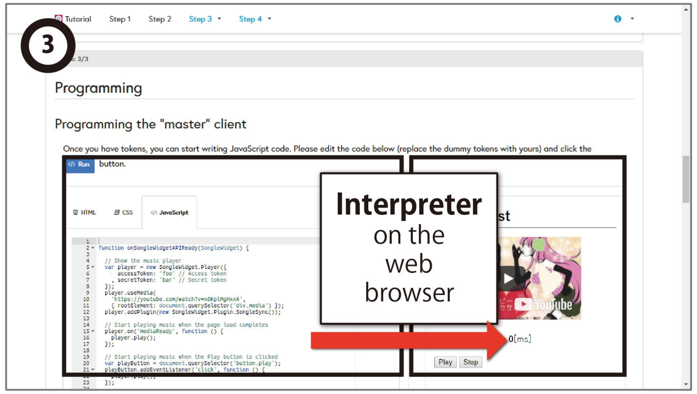


https://www.tutorialspoint.com/computer_programming/computer_programming_syntax.htm

```
Hello, World!
Hello, World!
```

```
main() {
   /* printf() function to write Hello, World! */
   printf( "Hello, World!" );
   printf( "Hello, World!" );
}
```

```
Compiling the source code....
$gcc main.c -o demo -lm -pthread -lgmp -lreadline 2>&1

Executing the program....
$demo
Hello, World!Hello, World!
```



Programming

Programming the "master" client

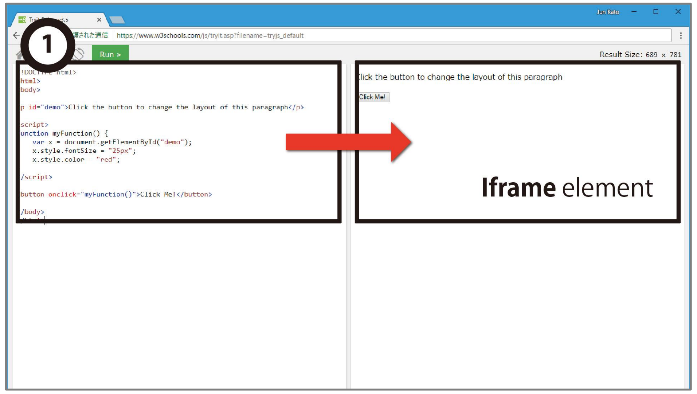
Once you have tokens, you can start writing JavaScript code. Please edit the code below (replace the dummy tokens with yours) and click the Run button.

```

function onSongleWidgetAPIReady(SongleWidget) {

  // Show the music player
  var player = new SongleWidget.Player({
      accessToken: 'foo' // Access token
    , secretToken: 'bar' // Secret token
  });
  player.useMedia(
    'https://youtube.com/watch?v=xOKplMgHxxA',
    { rootElement: document.querySelector('div.media') });
  player.addPlugin(new SongleWidget.Plugin.SongleSync());

  // Start playing music when the page load completes
  player.on('mediaReady', function () {
    player.play();
  });

  // Start playing music when the Play button is clicked
  var playButton = document.querySelector('button.play');
  playButton.addEventListener('click', function () {

  });
```

Toy:
cannot used for non-browser languages

Expensive:
requires lots of server-side resources

Toy sandbox OR expensive sandbox?

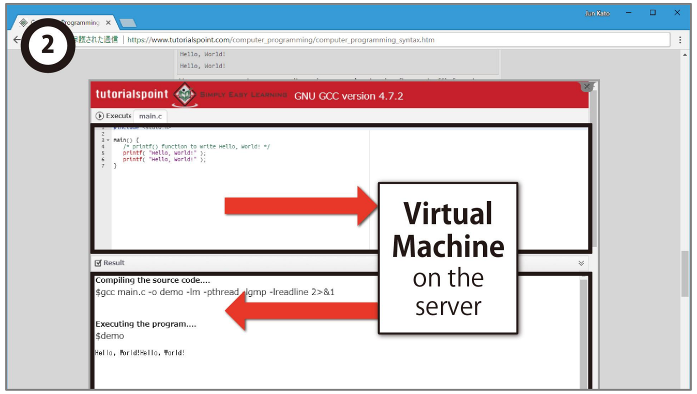


https://www.w3schools.com/js/tryit.asp?filename=tryjs_default

```
!DOCTYPE html>
html>
body>

p id="demo">Click the button to change the layout of this paragraph</p>

script>
unction myFunction() {
   var x = document.getElementById("demo");
   x.style.fontSize = "25px";
   x.style.color = "red";

/script>

button onclick="myFunction()">Click Me!</button>

/body>
```

lick the button to change the layout of this paragraph

Iframe element



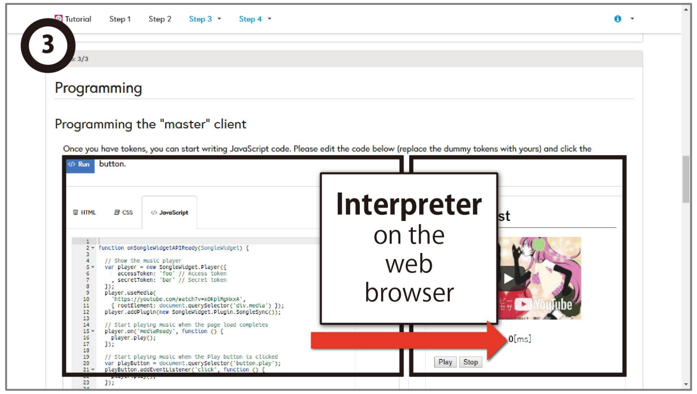
https://www.tutorialspoint.com/computer_programming/computer_programming_syntax.htm

GNU GCC version 4.7.2

```
main() {
   /* printf() function to write Hello, World! */
   printf( "Hello, World!" );
   printf( "Hello, World!" );
}
```

```
Compiling the source code....
$gcc main.c -o demo -lm -pthread -lgmp -lreadline 2>&1

Executing the program....
$demo
Hello, World!Hello, World!
```

Virtual Machine on the server



Programming

Programming the "master" client

Once you have tokens, you can start writing JavaScript code. Please edit the code below (replace the dummy tokens with yours) and click the Run button.

```
function onSongleWidgetAPIReady(SongleWidget) {

  // Show the music player
  var player = new SongleWidget.Player({
      accessToken: 'foo' // Access token
    , secretToken: 'bar' // Secret token
  });
  player.useMedia(
    'https://youtube.com/watch?v=xOKplMgHxxA',
    { rootElement: document.querySelector('div.media') });
  player.addPlugin(new SongleWidget.Plugin.SongleSync());

  // Start playing music when the page load completes
  player.on('mediaReady', function () {
    player.play();
  });

  // Start playing music when the Play button is clicked
  var playButton = document.querySelector('button.play');
  playButton.addEventListener('click', function () {
  });
```

Interpreter on the web browser

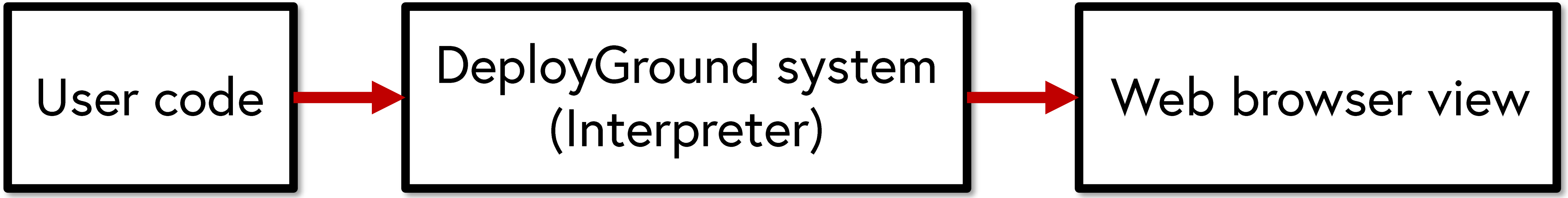
Toy:
cannot used for non-browser languages

Expensive:
requires lots of server-side resources

Flexible:
can potentially support any APIs

Pseudo runtime environment

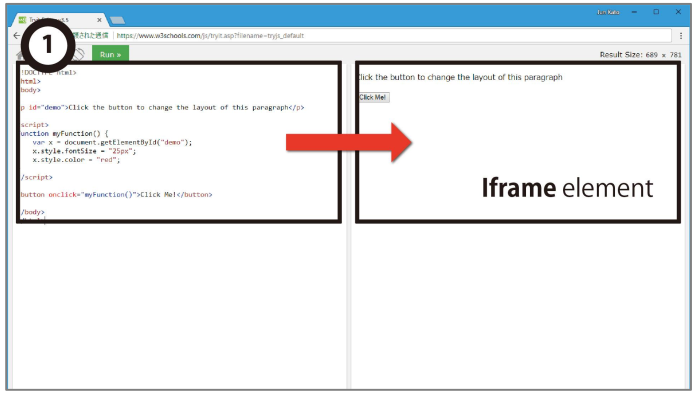
for making a flexible sandbox



- requires manual implementation of the emulation layer
- yet has potential to emulate anything that can be represented (visualized) on a web browser
- e.g.,
  - Node.js-based web servers: usually requires dedicated domain names
  - Physical computing devices: usually requires purchasing modules

LIMITATION (1)

Ephemeral code?

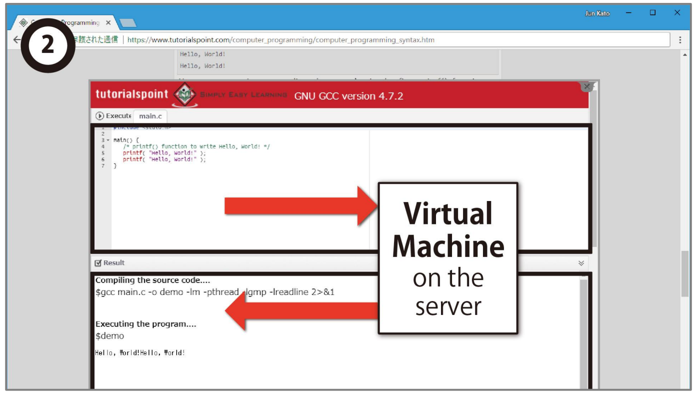


https://www.w3schools.com/js/tryit.asp?filename=tryjs_default

Result Size: 689 x 781

```
!DOCTYPE html>
html>
body>

p id="demo">Click the button to change the layout of this paragraph</p>

script>
unction myFunction() {
   var x = document.getElementById("demo");
   x.style.fontSize = "25px";
   x.style.color = "red";

/script>

button onclick="myFunction()">Click Me!</button>

/body>
```

lick the button to change the layout of this paragraph



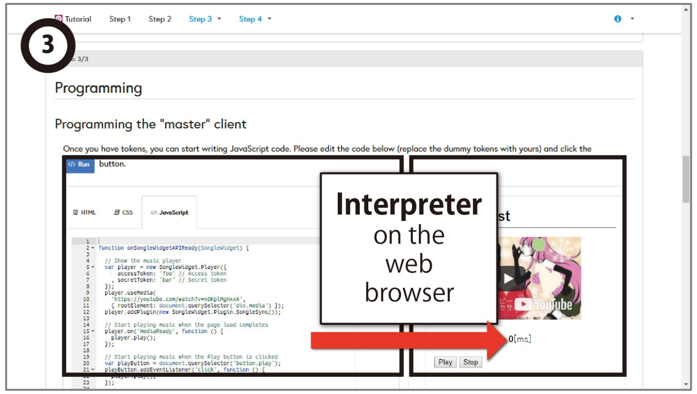
https://www.tutorialspoint.com/computer_programming/computer_programming_syntax.htm

```
Hello, World!
Hello, World!
```

GNU GCC version 4.7.2

```c
main() {
   /* printf() function to write Hello, World! */
   printf( "Hello, World!" );
   printf( "Hello, World!" );
}
```

```
Compiling the source code....
$gcc main.c -o demo -lm -pthread -lgmp -lreadline 2>&1

Executing the program....
$demo
Hello, World!Hello, World!
```



: 3/3

Programming

Programming the "master" client

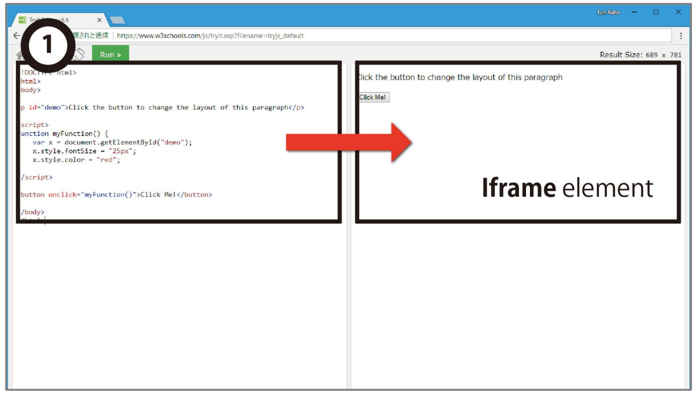
Once you have tokens, you can start writing JavaScript code. Please edit the code below (replace the dummy tokens with yours) and click the Run button.

```javascript

function onSongleWidgetAPIReady(SongleWidget) {

  // Show the music player
  var player = new SongleWidget.Player({
      accessToken: 'foo' // Access token
    , secretToken: 'bar' // Secret token
  });
  player.useMedia(
    'https://youtube.com/watch?v=xOKplMgHxxA',
    { rootElement: document.querySelector('div.media') });
  player.addPlugin(new SongleWidget.Plugin.SongleSync());

  // Start playing music when the page load completes
  player.on('mediaReady', function () {
    player.play();
  });

  // Start playing music when the Play button is clicked
  var playButton = document.querySelector('button.play');
  playButton.addEventListener('click', function () {

  });
```

Download each code editor content as a file

VM sessions cannot be (easily) exported

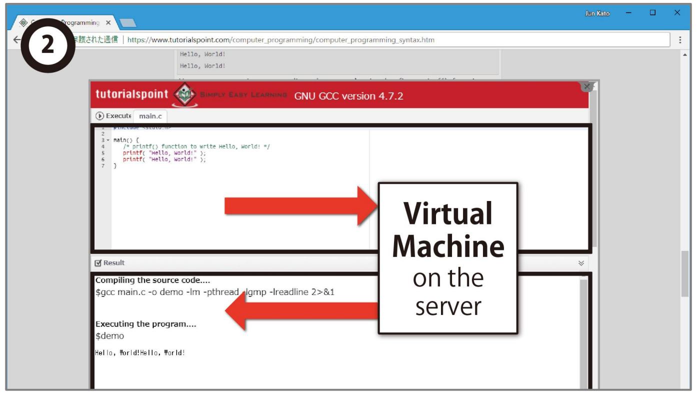
Ephemeral code?



https://www.w3schools.com/js/tryit.asp?filename=tryjs_default

```
!DOCTYPE html>
html>
body>

p id="demo">Click the button to change the layout of this paragraph</p>

script>
unction myFunction() {
   var x = document.getElementById("demo");
   x.style.fontSize = "25px";
   x.style.color = "red";

/script>

button onclick="myFunction()">Click Me!</button>

/body>
```

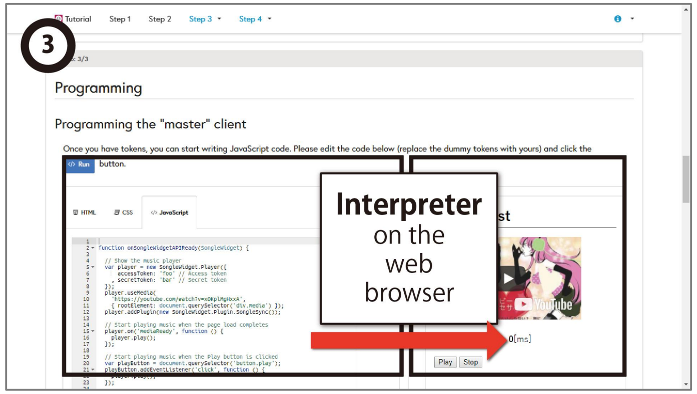
lick the button to change the layout of this paragraph



https://www.tutorialspoint.com/computer_programming/computer_programming_syntax.htm

```
main() {
   /* printf() function to write Hello, World! */
   printf( "Hello, World!" );
   printf( "Hello, World!" );
}
```

```
Compiling the source code....
$gcc main.c -o demo -lm -pthread -lgmp -lreadline 2>&1

Executing the program....
$demo
Hello, World!Hello, World!
```



Programming

Programming the "master" client

Once you have tokens, you can start writing JavaScript code. Please edit the code below (replace the dummy tokens with yours) and click the Run button.

```
function onSongleWidgetAPIReady(SongleWidget) {

  // Show the music player
  var player = new SongleWidget.Player({
    accessToken: 'foo' // Access token
    , secretToken: 'bar' // Secret token
  });
  player.useMedia(
    'https://youtube.com/watch?v=xOKplMgHxxA',
    { rootElement: document.querySelector('div.media') });
  player.addPlugin(new SongleWidget.Plugin.SongleSync());

  // Start playing music when the page load completes
  player.on('mediaReady', function () {
    player.play();
  });

  // Start playing music when the Play button is clicked
  var playButton = document.querySelector('button.play');
  playButton.addEventListener('click', function () {
  });
```

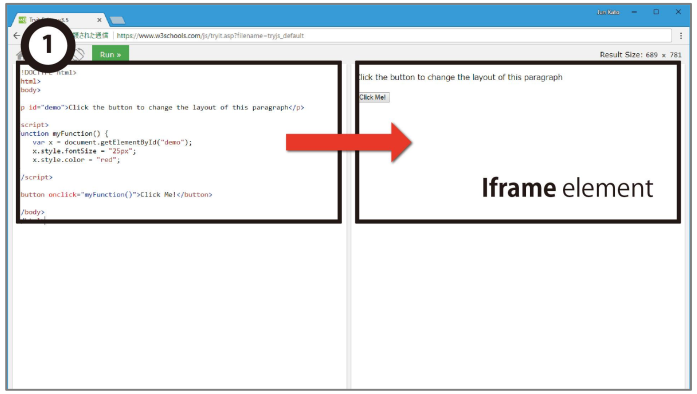
Download each code editor content as a file

VM sessions cannot be (easily) exported

Save the entire workspace as a GitHub repo

LIMITATION (3)

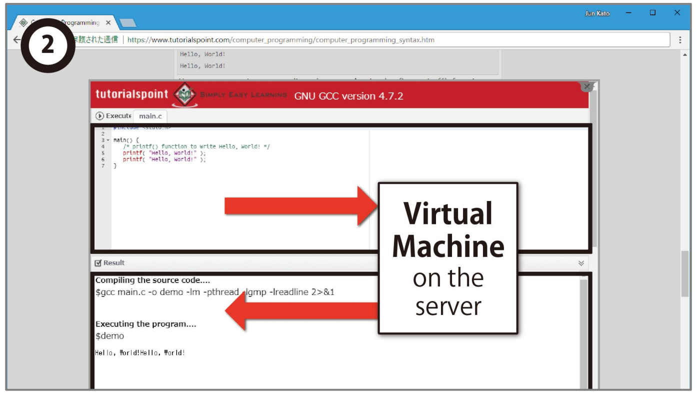
No support for deployment?



```
!DOCTYPE html>
html>
body>

p id="demo">Click the button to change the layout of this paragraph</p>

script>
unction myFunction() {
  var x = document.getElementById("demo");
  x.style.fontSize = "25px";
  x.style.color = "red";

/script>

button onclick="myFunction()">Click Me!</button>

/body>
```

lick the button to change the layout of this paragraph

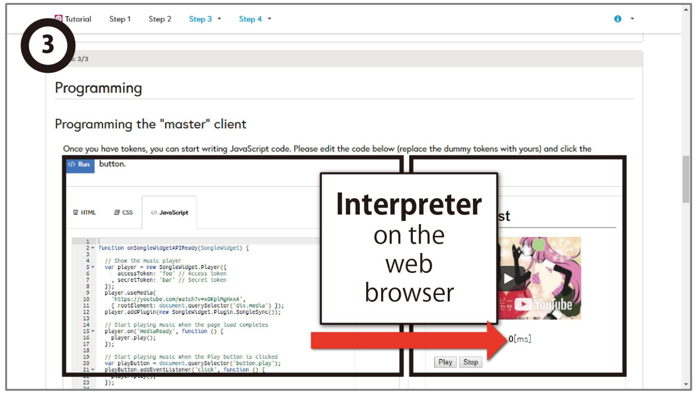
Click Me!


2
Programming
Jun Kato
https://www.tutorialspoint.com/computer_programming/computer_programming_syntax.htm
tutorialspoint
SIMPLY EASY LEARNING
GNU GCC version 4.7.2
Execute
main.c
Result
Virtual Machine on the server

```
Hello, World!
Hello, World!
```

```
main() {
   /* printf() function to write Hello, World! */
   printf( "Hello, World!" );
   printf( "Hello, World!" );
}
```

```
Compiling the source code....
$gcc main.c -o demo -lm -pthread -lgmp -lreadline 2>&1

Executing the program....
$demo
Hello, World!Hello, World!
```



Programming

Programming the "master" client

Once you have tokens, you can start writing JavaScript code. Please edit the code below (replace the dummy tokens with yours) and click the Run button.

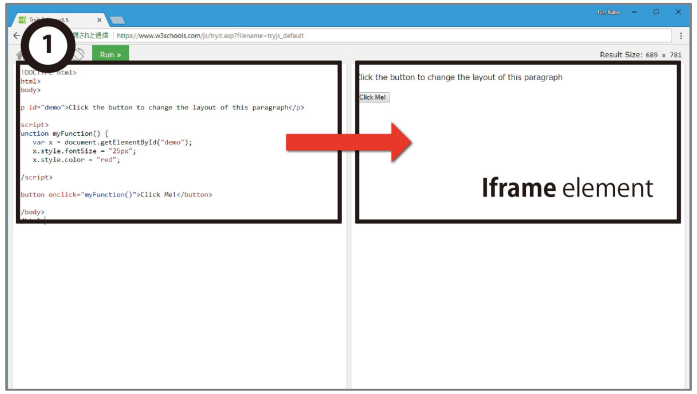
```
function onSongleWidgetAPIReady(SongleWidget) {

  // Show the music player
  var player = new SongleWidget.Player({
      accessToken: 'foo' // Access token
    , secretToken: 'bar' // Secret token
  });
  player.useMedia(
    'https://youtube.com/watch?v=xOKplMgHxxA',
    { rootElement: document.querySelector('div.media') });
  player.addPlugin(new SongleWidget.Plugin.SongleSync());

  // Start playing music when the page load completes
  player.on('mediaReady', function () {
    player.play();
  });

  // Start playing music when the Play button is clicked
  var playButton = document.querySelector('button.play');
  playButton.addEventListener('click', function () {
  });
```

HTML with JS files cannot be directly opened with modern web browsers

In more complex cases (e.g. Node.js-based projects), only instructions are presented

LIMITATION (3)

No support for deployment?



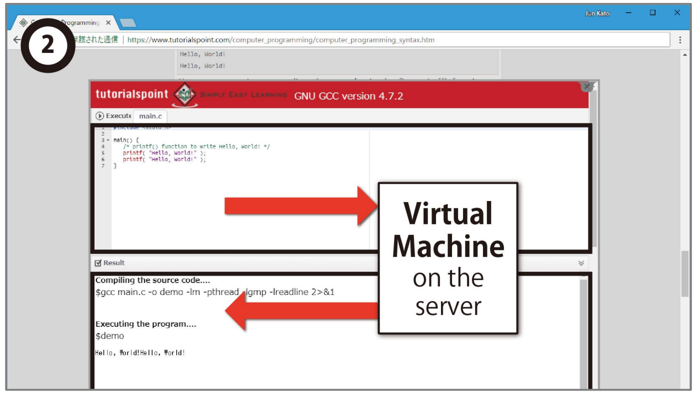
https://www.w3schools.com/js/tryit.asp?filename=tryjs_default

Result Size: 689 x 781

```
!DOCTYPE html>
html>
body>

p id="demo">Click the button to change the layout of this paragraph</p>

script>
unction myFunction() {
   var x = document.getElementById("demo");
   x.style.fontSize = "25px";
   x.style.color = "red";

/script>

button onclick="myFunction()">Click Me!</button>

/body>
```

lick the button to change the layout of this paragraph

Iframe element



https://www.tutorialspoint.com/computer_programming/computer_programming_syntax.htm

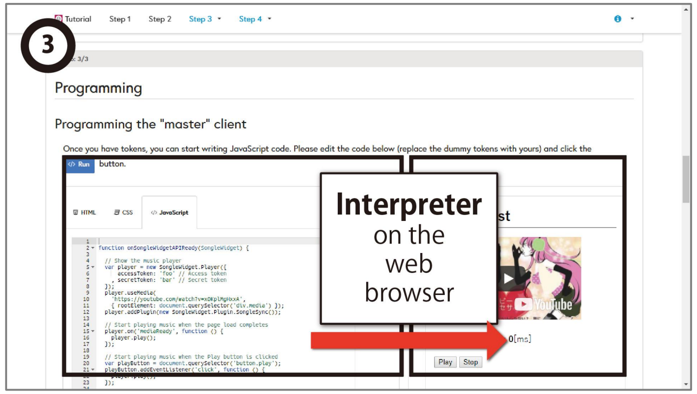
```
Hello, World!
Hello, World!
```

GNU GCC version 4.7.2

```
main() {
   /* printf() function to write Hello, World! */
   printf( "Hello, World!" );
   printf( "Hello, World!" );
}
```

```
Compiling the source code....
$gcc main.c -o demo -lm -pthread -lgmp -lreadline 2>&1

Executing the program....
$demo
Hello, World!Hello, World!
```

Virtual Machine on the server



: 3/3

Programming

Programming the "master" client

Once you have tokens, you can start writing JavaScript code. Please edit the code below (replace the dummy tokens with yours) and click the Run button.

```
function onSongleWidgetAPIReady(SongleWidget) {

  // Show the music player
  var player = new SongleWidget.Player({
      accessToken: 'foo' // Access token
    , secretToken: 'bar' // Secret token
  });
  player.useMedia(
    'https://youtube.com/watch?v=xOKplMgHxxA',
    { rootElement: document.querySelector('div.media') });
  player.addPlugin(new SongleWidget.Plugin.SongleSync());

  // Start playing music when the page load completes
  player.on('mediaReady', function () {
    player.play();
  });

  // Start playing music when the Play button is clicked
  var playButton = document.querySelector('button.play');
  playButton.addEventListener('click', function () {

  });
```

0[ms]

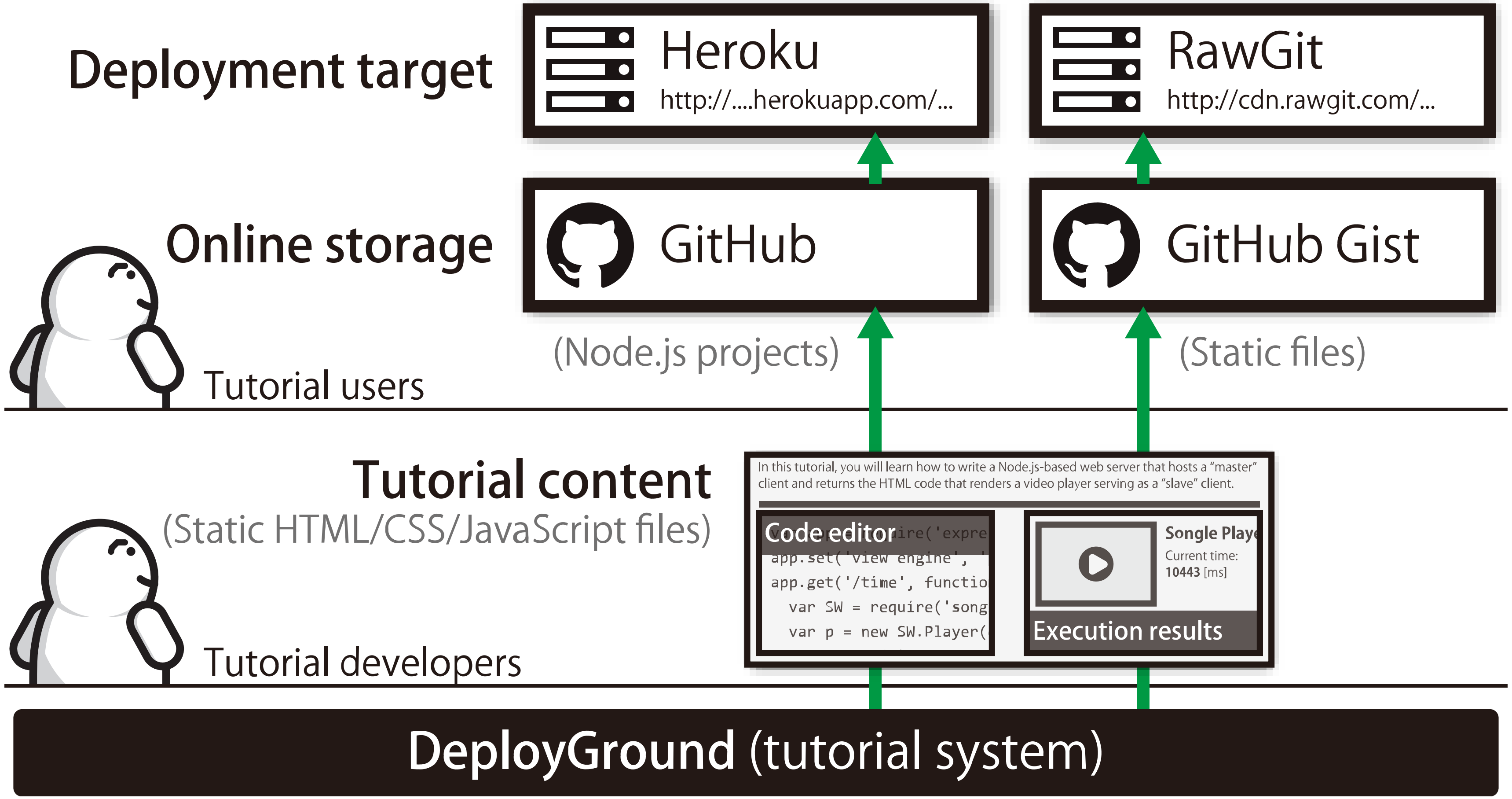
Interpreter on the web browser

HTML with JS files cannot be directly opened with modern web browsers

In more complex cases (e.g. Node.js-based projects), only instructions are presented

Deploy to GitHub Gist or Heroku

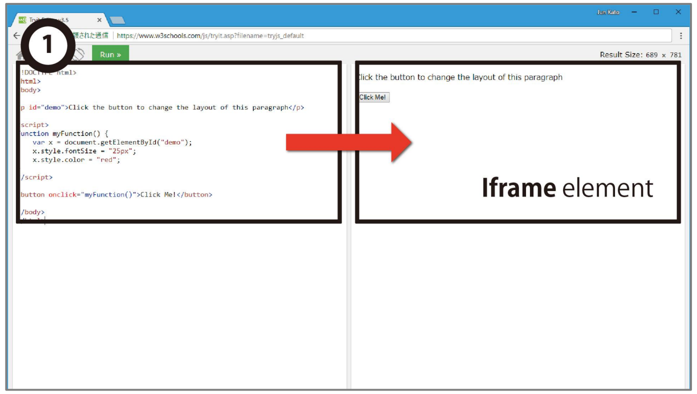
Adaptive boilerplate
for exporting files as an executable project



In this tutorial, you will learn how to write a Node.js-based web server that hosts a “master” client and returns the HTML code that renders a video player serving as a “slave” client.

```
var app = require('expre
app.set('view engine', '
app.get('/time', functio
  var SW = require('song
  var p = new SW.Player(
```

Little social interaction?

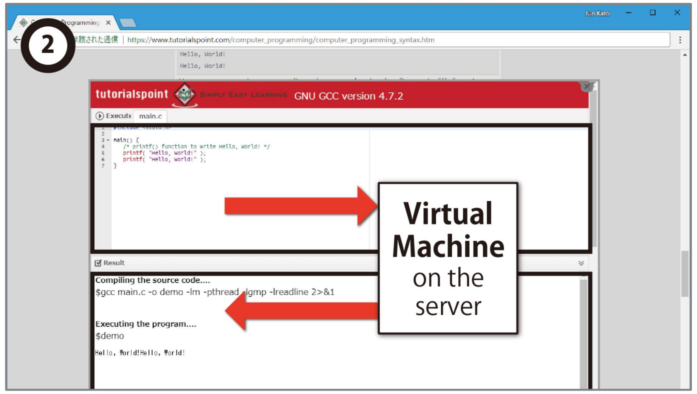


https://www.w3schools.com/js/tryit.asp?filename=tryjs_default

Result Size: 689 x 781

```
!DOCTYPE html>
html>
body>

p id="demo">Click the button to change the layout of this paragraph</p>

script>
unction myFunction() {
   var x = document.getElementById("demo");
   x.style.fontSize = "25px";
   x.style.color = "red";

/script>

button onclick="myFunction()">Click Me!</button>

/body>
```

lick the button to change the layout of this paragraph

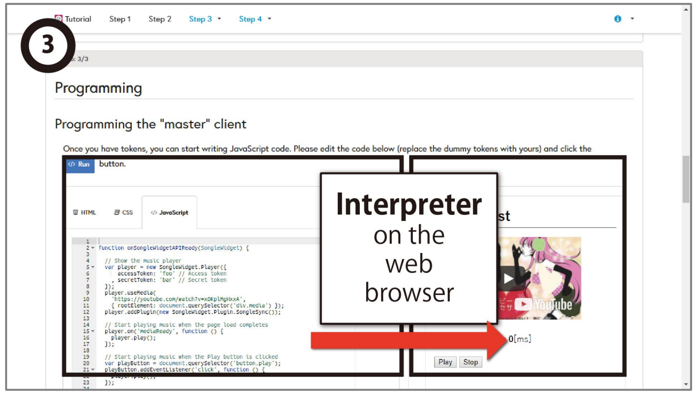


https://www.tutorialspoint.com/computer_programming/computer_programming_syntax.htm

GNU GCC version 4.7.2

```
main() {
   /* printf() function to write Hello, World! */
   printf( "Hello, World!" );
   printf( "Hello, World!" );
}
```

```
Compiling the source code....
$gcc main.c -o demo -lm -pthread -lgmp -lreadline 2>&1

Executing the program....
$demo
Hello, World!Hello, World!
```



Programming

Programming the "master" client

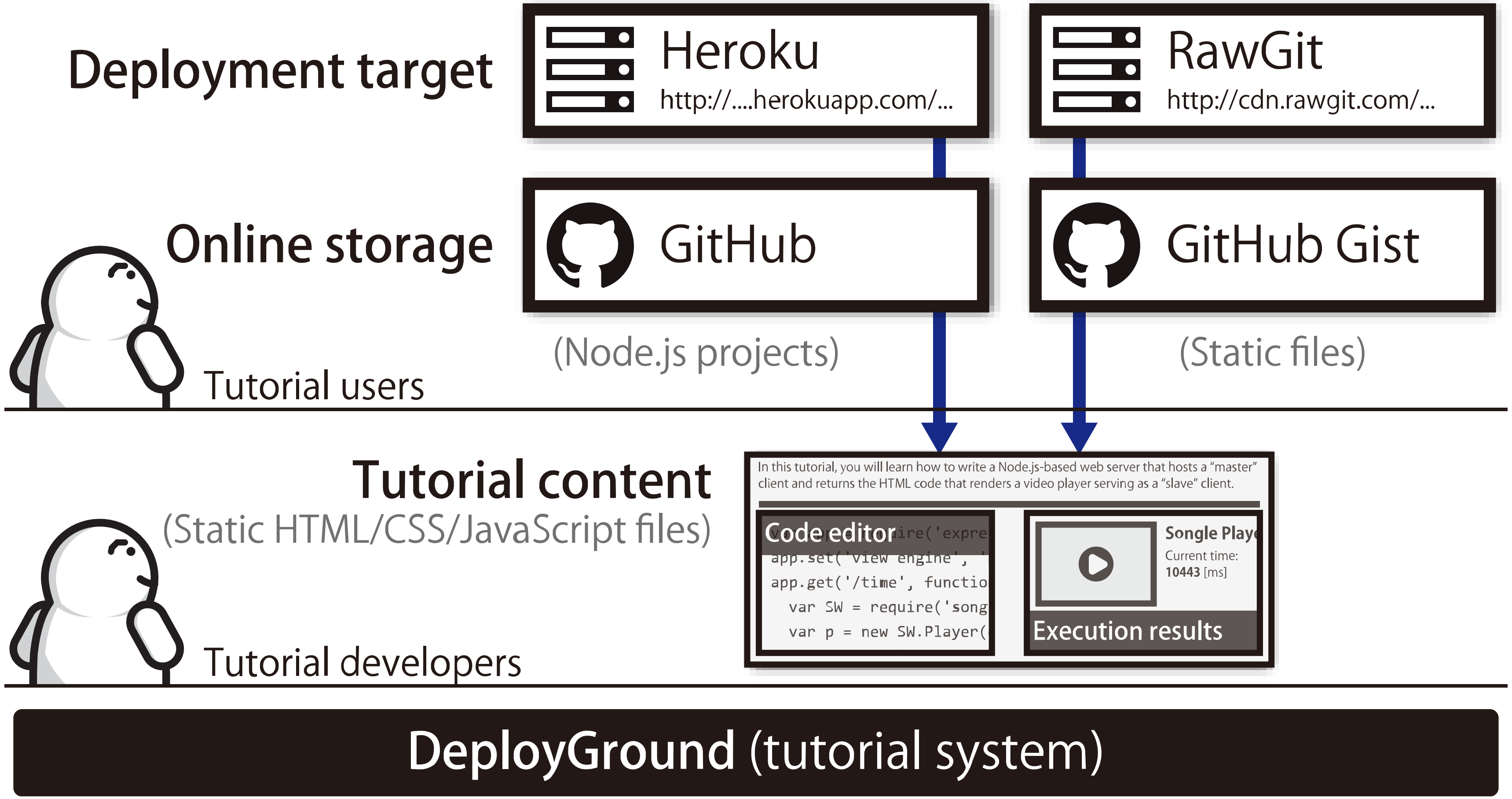
Once you have tokens, you can start writing JavaScript code. Please edit the code below (replace the dummy tokens with yours) and click the Run button.

```
function onSongleWidgetAPIReady(SongleWidget) {

  // Show the music player
  var player = new SongleWidget.Player({
    accessToken: 'foo' // Access token
  , secretToken: 'bar' // Secret token
  });
  player.useMedia(
    'https://youtube.com/watch?v=xOKplMgHxxA',
    { rootElement: document.querySelector('div.media') });
  player.addPlugin(new SongleWidget.Plugin.SongleSync());

  // Start playing music when the page load completes
  player.on('mediaReady', function () {
    player.play();
  });

  // Start playing music when the Play button is clicked
  var playButton = document.querySelector('button.play');
  playButton.addEventListener('click', function () {

  });
```

"Achievement unlocked!" kind of posts can be potentially made on social networking services

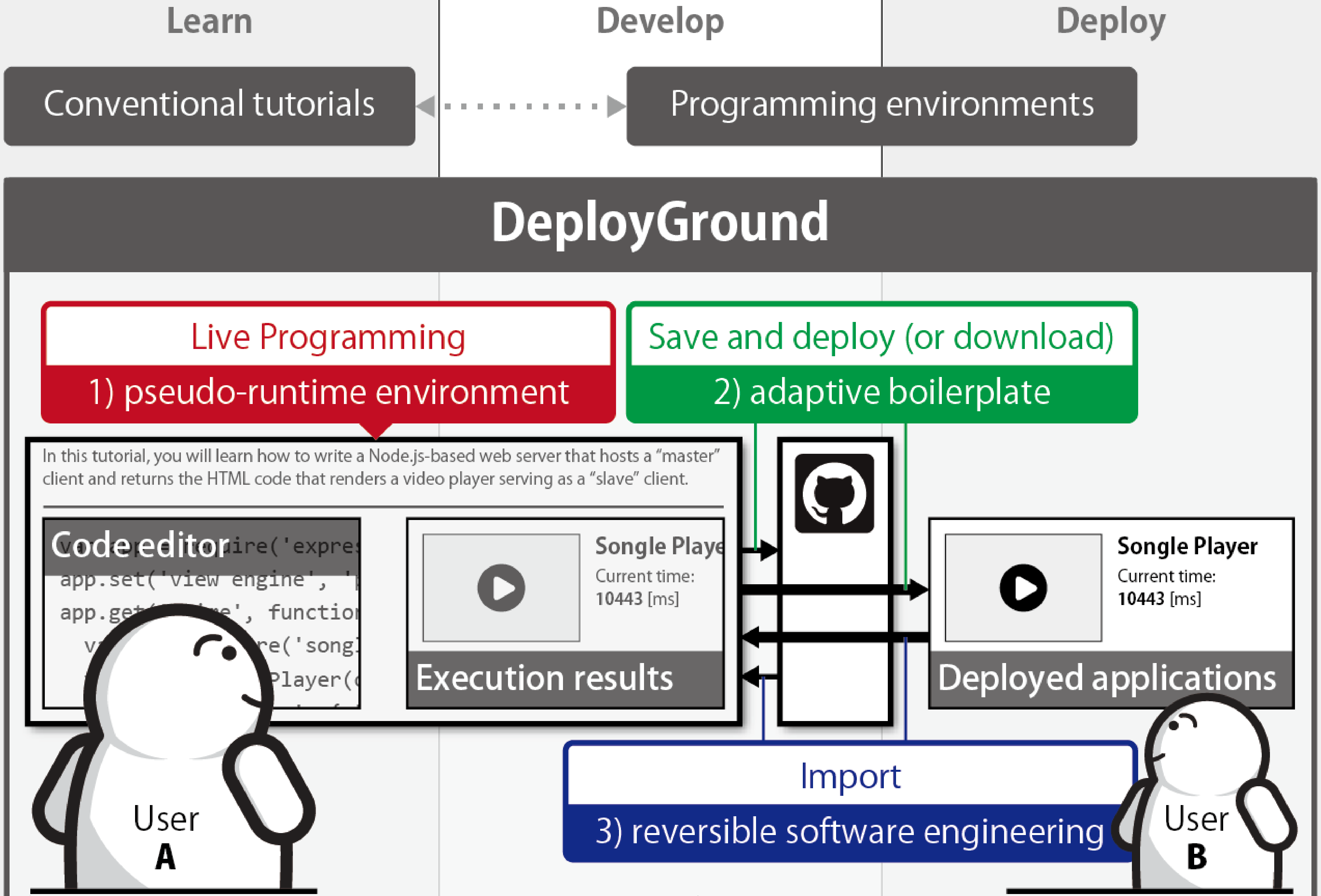
Codebases and apps can be shared instantly

Reversible software engineering
to use people's outcome as educational resource



In this tutorial, you will learn how to write a Node.js-based web server that hosts a “master” client and returns the HTML code that renders a video player serving as a “slave” client.

```
var app = require('expre
app.set('view engine', '
app.get('/time', functio
  var SW = require('song
  var p = new SW.Player(
```



Learn

Develop

Deploy

Conventional tutorials

Programming environments

DeployGround

Live Programming

1) pseudo-runtime environment

Save and deploy (or download)

2) adaptive boilerplate

In this tutorial, you will learn how to write a Node.js-based web server that hosts a “master” client and returns the HTML code that renders a video player serving as a “slave” client.

Code editor

```
app.set('view engine', 'p
app.get(   re', functio
  v       re('song
          Player(c
```

Songle Playe

Current time:
10443 [ms]

Execution results

Songle Player

Current time:
10443 [ms]

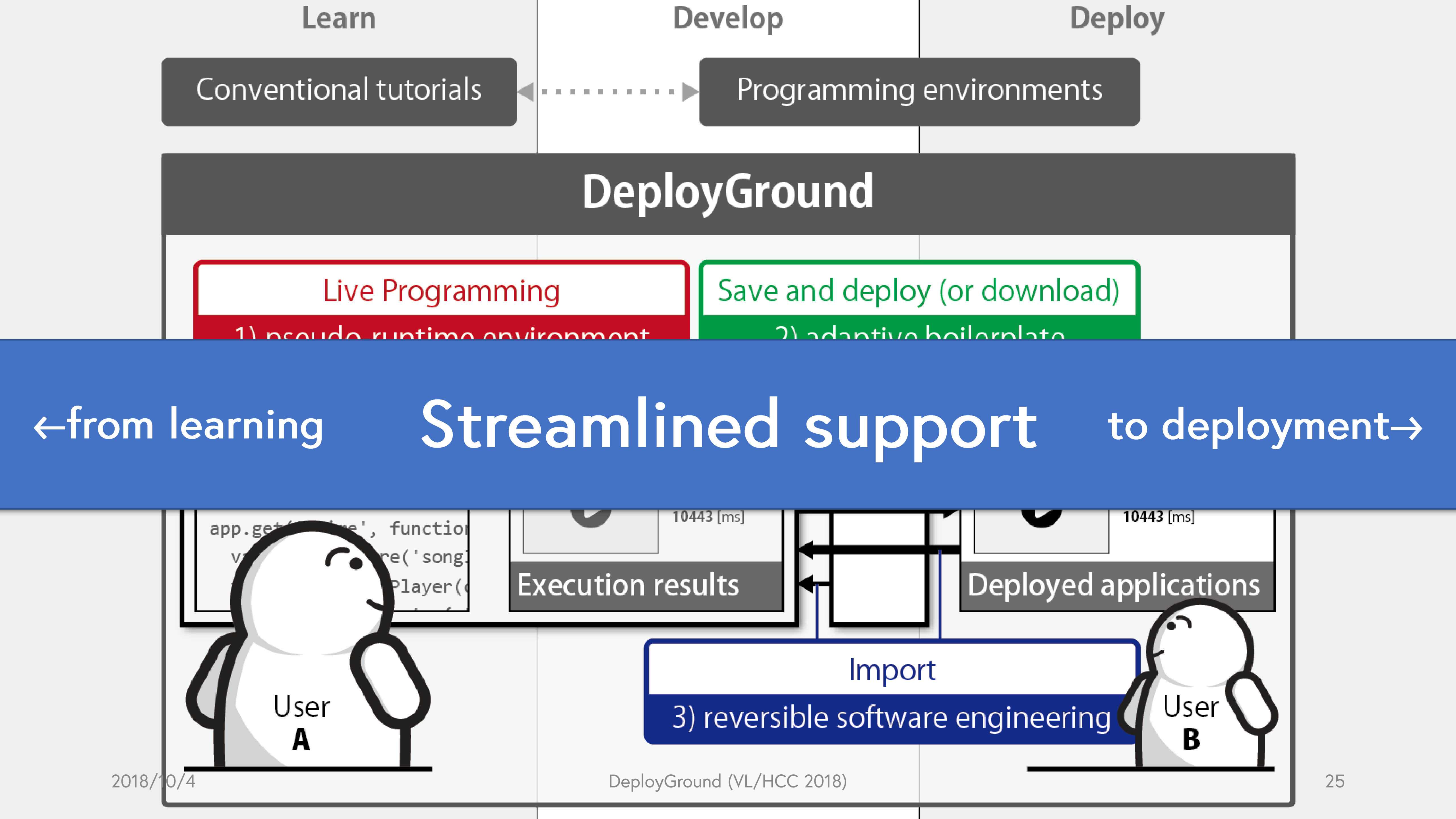
Deployed applications

Import

3) reversible software engineering

User
A

User
B



Learn

Develop

Deploy

Conventional tutorials

Programming environments

DeployGround

Live Programming

1) pseudo-runtime environment

Save and deploy (or download)

2) adaptive boilerplate

←from learning Streamlined support to deployment→

Execution results

Deployed applications

Import

3) reversible software engineering

Preliminary user feedback

Refer to the paper

Asked 3 software engineers and 2 researchers to answer prequestionnaire and try out the tutorial

Asked 24 university students to form 6 groups and prototype applications in 2 days

- All of them successfully benefited from the framework
- Potential applications: APIs with expensive initial cost to try out
- Requests for more detailed views on save/deploy features
- Emulation is imperfect; more diverse examples are demanded

SUMMARY (PROPOSED FRAMEWORK):

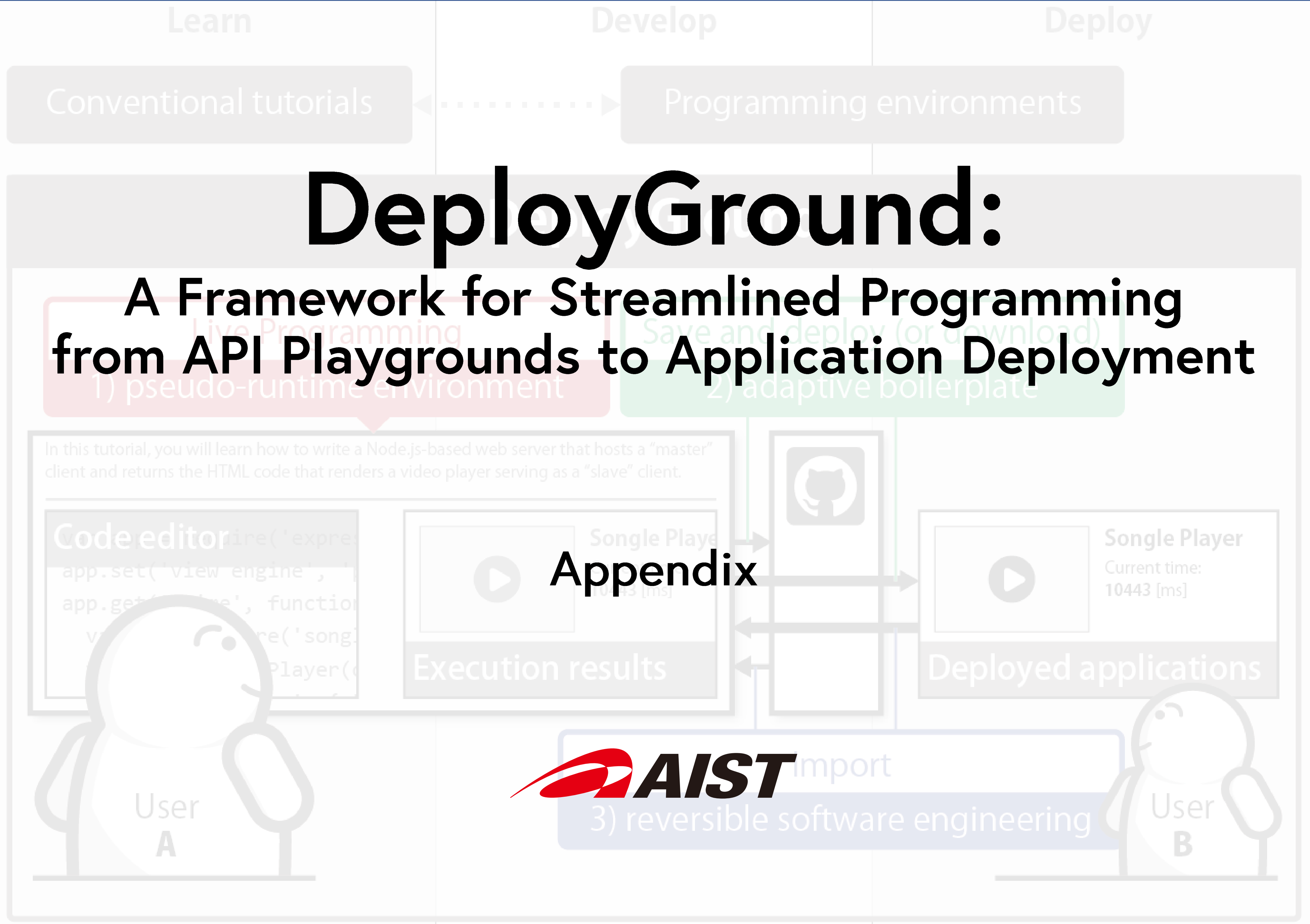
DeployGround framework

- Pseudo runtime environment enables live programming experience.
  Learners can enjoy testing the APIs. They accumulate their codebase during the tutorial.
- Adaptive boilerplate brings the experience out of the sandbox.
  In the end, they get the archived project files. The tutorial even helps them deploy the project.
- Reversible software engineering allows to gain benefits from social coding.
  All of the above experiences can be easily shared on the web, helping the other learners.

DeployGround:

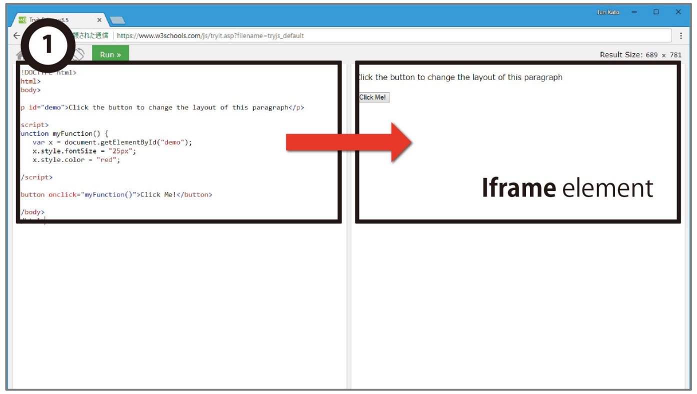
A Framework for Streamlined Programming from API Playgrounds to Application Deployment

Appendix



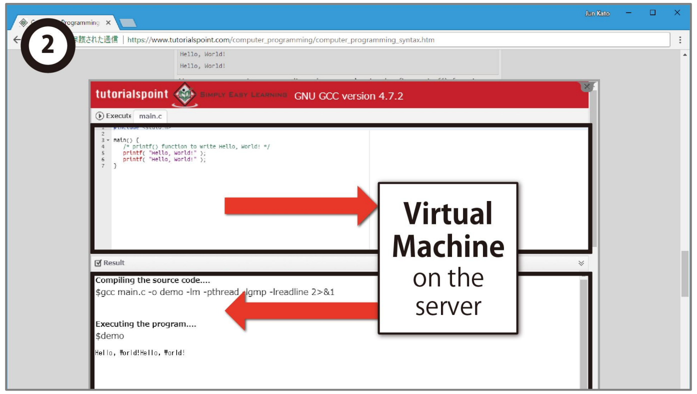
In this tutorial, you will learn how to write a Node.js-based web server that hosts a "master" client and returns the HTML code that renders a video player serving as a "slave" client.

"Playground" implementations


1
Run »
https://www.w3schools.com/js/tryit.asp?filename=tryjs_default
Result Size: 689 x 781
Click Me!
Iframe element

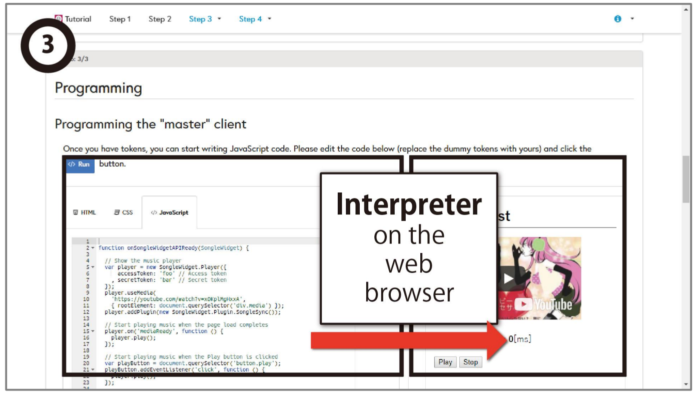
```
!DOCTYPE html>
html>
body>

p id="demo">Click the button to change the layout of this paragraph</p>

script>
unction myFunction() {
   var x = document.getElementById("demo");
   x.style.fontSize = "25px";
   x.style.color = "red";

/script>

button onclick="myFunction()">Click Me!</button>

/body>
```

lick the button to change the layout of this paragraph


2
https://www.tutorialspoint.com/computer_programming/computer_programming_syntax.htm
tutorialspoint
SIMPLY EASY LEARNING
GNU GCC version 4.7.2
Execute
main.c
Result
Virtual Machine on the server

```
main() {
   /* printf() function to write Hello, World! */
   printf( "Hello, World!" );
   printf( "Hello, World!" );
}
```

```
Compiling the source code....
$gcc main.c -o demo -lm -pthread -lgmp -lreadline 2>&1


Executing the program....
$demo

Hello, World!Hello, World!
```



Programming

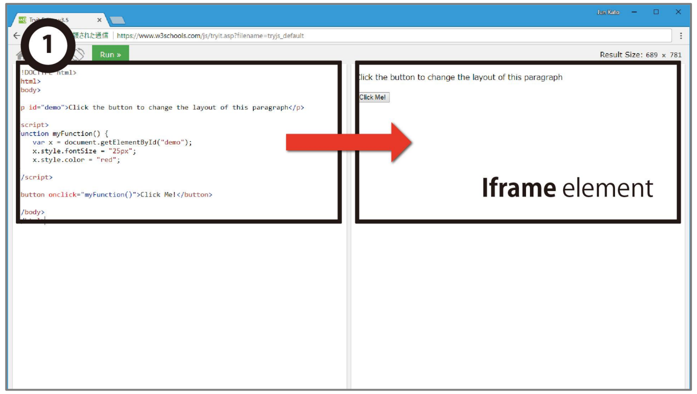
Programming the "master" client

Once you have tokens, you can start writing JavaScript code. Please edit the code below (replace the dummy tokens with yours) and click the Run button.

```

function onSongleWidgetAPIReady(SongleWidget) {

  // Show the music player
  var player = new SongleWidget.Player({
      accessToken: 'foo' // Access token
    , secretToken: 'bar' // Secret token
  });
  player.useMedia(
    'https://youtube.com/watch?v=xOKplMgHxxA',
    { rootElement: document.querySelector('div.media') });
  player.addPlugin(new SongleWidget.Plugin.SongleSync());

  // Start playing music when the page load completes
  player.on('mediaReady', function () {
    player.play();
  });

  // Start playing music when the Play button is clicked
  var playButton = document.querySelector('button.play');
  playButton.addEventListener('click', function () {

  });
```

Performance		
Browser-native	Huge latency	A little overhead
Debuggability		
Very low	Low	High
Supported APIs:		
Complete (browser-based APIs)	Complete (CUI-based APIs)	Partial (any emulatable APIs)

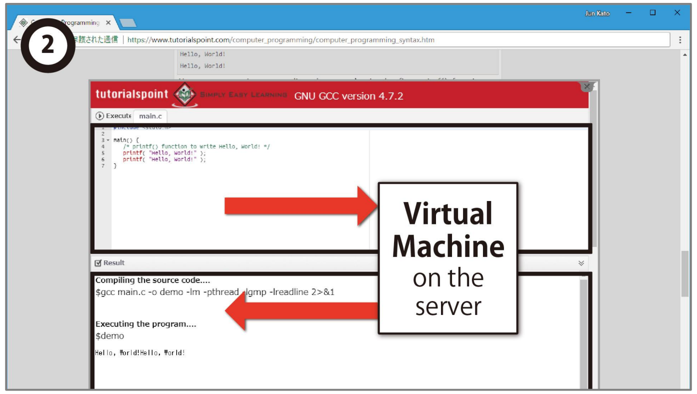
"Playground" implementations


1
Tryit Editor v3.5
保護された通信 | https://www.w3schools.com/js/tryit.asp?filename=tryjs_default
Run »
Result Size: 689 x 781
Jun Kato
Iframe element

```
!DOCTYPE html>
html>
body>

p id="demo">Click the button to change the layout of this paragraph</p>

script>
unction myFunction() {
   var x = document.getElementById("demo");
   x.style.fontSize = "25px";
   x.style.color = "red";

/script>

button onclick="myFunction()">Click Me!</button>

/body>
```

Click the button to change the layout of this paragraph

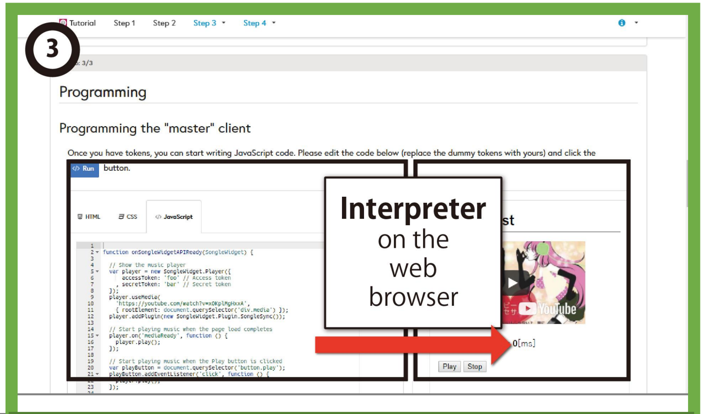
Click Me!


2
Programming
保護された通信 | https://www.tutorialspoint.com/computer_programming/computer_programming_syntax.htm
Jun Kato
tutorialspoint
Simply Easy Learning
GNU GCC version 4.7.2
Execute
main.c
Result
Virtual Machine on the server

```
Hello, World!
Hello, World!
```

```
main() {
    /* printf() function to write Hello, World! */
    printf( "Hello, World!" );
    printf( "Hello, World!" );
}
```

```
Compiling the source code....
$gcc main.c -o demo -lm -pthread -lgmp -lreadline 2>&1

Executing the program....
$demo
Hello, World!Hello, World!
```



Programming

Programming the "master" client

Once you have tokens, you can start writing JavaScript code. Please edit the code below (replace the dummy tokens with yours) and click the Run button.

```
function onSongleWidgetAPIReady(SongleWidget) {

  // Show the music player
  var player = new SongleWidget.Player({
      accessToken: 'foo' // Access token
    , secretToken: 'bar' // Secret token
  });
  player.useMedia(
    'https://youtube.com/watch?v=xOKplMgHxxA',
    { rootElement: document.querySelector('div.media') });
  player.addPlugin(new SongleWidget.Plugin.SongleSync());

  // Start playing music when the page load completes
  player.on('mediaReady', function () {
    player.play();
  });

  // Start playing music when the Play button is clicked
  var playButton = document.querySelector('button.play');
  playButton.addEventListener('click', function () {

  });
```

Performance		DeployGround
Browser-native	Huge latency	
Debuggability		
Very low	Low	
Supported APIs:		
Complete (browser-based APIs)	Complete (CUI-based APIs)	