

## Global Execution Context

### Global Memory

Identifier	Value

Current Operation: **Downloading Javascript source code**

## Global Execution Context

## CREATION PHASE

```
var name = "Luigi";  
let input = "Hello, World!";  
  
function broadcast(message) {  
  return `${name} says ${message}`;  
}  
  
console.log(broadcast(input));
```

Global Memory	
Identifier	Value
window	<Global Object>
this	<window Object>

Current Operation: **Program**

## Global Execution Context

## CREATION PHASE

```
→ var name = "Luigi";  
  let input = "Hello, World!";  
  
  function broadcast(message) {  
    return `${name} says ${message}`;  
  }  
  
  console.log(broadcast(input));
```

Global Memory	
Identifier	Value
window	<Global Object>
this	<window Object>
name	undefined

Current Operation: **Variable Declaration**

## Global Execution Context

## CREATION PHASE

```
var name = "Luigi";  
let input = "Hello, World!";  
  
function broadcast(message) {  
  return `${name} says ${message}`;  
}  
  
console.log(broadcast(input));
```

### Global Memory

Identifier	Value
window	<Global Object>
this	<window Object>
name	undefined
input	<uninitialized>

Current Operation: **Variable Declaration**

## Global Execution Context

## CREATION PHASE

```
var name = "Luigi";  
let input = "Hello, World!";
```



```
function broadcast(message) {  
  return `${name} says ${message}`;  
}
```

```
console.log(broadcast(input));
```

### Global Memory

Identifier	Value
window	<Global Object>
this	<window Object>
name	undefined
input	<uninitialized>
<b>broadcast</b>	<fn object>

Current Operation: **Function Declaration**

## Global Execution Context

## EXECUTION PHASE

```
var name = "Luigi";  
let input = "Hello, World!";  
  
function broadcast(message) {  
  return `${name} says ${message}`;  
}  
  
console.log(broadcast(input));
```

### Global Memory

Identifier	Value
window	<Global Object>
this	<window Object>
name	"Luigi"
input	<uninitialized>
broadcast	<fn object>

Current Operation: **Variable Assignment**

## Global Execution Context

## EXECUTION PHASE

```
var name = "Luigi";  
let input = "Hello, World!";  
  
function broadcast(message) {  
  return `${name} says ${message}`;  
}  
  
console.log(broadcast(input));
```

### Global Memory

Identifier	Value
window	<Global Object>
this	<window Object>
name	"Luigi"
input	"Hello, World!"
broadcast	<fn object>

Current Operation: **Variable Assignment**

## Global Execution Context

## EXECUTION PHASE

```
var name = "Luigi";  
let input = "Hello, World!";  
  
function broadcast(message) {  
  return `${name} says ${message}`;  
}
```

➔ console.log(broadcast(input));

### Global Memory

Identifier	Value
window	<Global Object>
this	<window Object>
name	"Luigi"
input	"Hello, World!"
<b>broadcast</b>	<fn object>

Current Operation: **Call Expression**



## Global Execution Context

## EXECUTION PHASE

```
var name = "Luigi";  
let input = "Hello, World!";  
  
function broadcast(message) {  
  return `${name} says ${message}`.  
}
```

Global Memory	
Identifier	Value
window	<Global Object>
this	<window Object>

→ console.log(broadcast(input));

## Local Execution Context

## CREATION PHASE

```
function broadcast(message) {  
  return `${name} says ${message}`;  
}
```

Local Memory	
Identifier	Value
arguments	{length: 0}
this	<window Obj>

Current Operation: **Function Expression**

## Global Execution Context

## EXECUTION PHASE

```
var name = "Luigi";  
let input = "Hello, World!";  
  
function broadcast(message) {  
  return `${name} says ${message}`.  
}
```

Global Memory	
Identifier	Value
window	<Global Object>
this	<window Object>

→ console.log(broadcast(input));

## Local Execution Context

## CREATION PHASE

```
function broadcast(message) {  
  return `${name} says ${message}`;  
}
```

Local Memory	
Identifier	Value
arguments	{ 0: message, length: 1 }
this	<window Obj>

Current Operation: **Function Expression**

## Global Execution Context

## EXECUTION PHASE

```
var name = "Luigi";  
let input = "Hello, World!";  
  
function broadcast(message) {  
  return `${name} says ${message}`;  
}
```

Global Memory	
Identifier	Value
window	<Global Object>
this	<window Object>

→ console.log(broadcast(input));

## Local Execution Context

## CREATION PHASE

```
function broadcast(message) {  
  return `${name} says ${message}`;  
}
```

Local Memory	
Identifier	Value
arguments	{ 0: message, length: 1 }
this	<window Obj>
message	undefined

Current Operation: **Variable Declaration**

## Global Execution Context

EXECUTION PHASE

```
var name = "Luigi";  
let input = "Hello, World!";  
  
function broadcast(message) {  
  return `${name} says ${message}`;  
}
```

Global Memory	
Identifier	Value
window	<Global Object>
this	<window Object>

→ console.log(broadcast(input));

## Local Execution Context

EXECUTION PHASE

```
function broadcast(message) {  
  return `${name} says ${message}`;  
}
```

Local Memory	
Identifier	Value
arguments	{ 0: message, length: 1 }
this	<window Obj>
message	"Hello, World!"

Current Operation: **Variable Assignment**

## Global Execution Context

EXECUTION PHASE

```
var name = "Luigi";  
let input = "Hello, World!";  
  
function broadcast(message) {  
  return `${name} says ${message}`;  
}
```

Global Memory	
Identifier	Value
window	<Global Object>
this	<window Object>

→ console.log(broadcast(input));

## Local Execution Context

EXECUTION PHASE

→ function broadcast(message) {  
 return `\${name} says \${message}`;  
}

Local Memory	
Identifier	Value
arguments	{ 0: message, length: 1 }
this	<window Obj>
message	"Hello, World!"

Current Operation: **Block Statement**

## Global Execution Context

EXECUTION PHASE

```
var name = "Luigi";  
let input = "Hello, World!";  
  
function broadcast(message) {  
  return `${name} says ${message}`;  
}
```

Global Memory	
Identifier	Value
window	<Global Object>
this	<window Object>

→ console.log(broadcast(input));

## Local Execution Context

EXECUTION PHASE

→ function broadcast(message) {  
 return `\${name} says \${message}`;  
}

Local Memory	
Identifier	Value
arguments	{ 0: message, length: 1 }
this	<window Obj>
message	"Hello, World!"

Current Operation: **Return Statement**

## Global Execution Context

EXECUTION PHASE

```
var name = "Luigi";  
let input = "Hello, World!";  
  
function broadcast(message) {  
  return `${name} says ${message}`.  
}
```

Global Memory	
Identifier	Value
window	<Global Object>
this	<window Object>

→ console.log(broadcast(input));

## Local Execution Context

EXECUTION PHASE

→ function broadcast(message) {  
 return `\${name} says \${message}`;  
}

Local Memory	
Identifier	Value
arguments	{ 0: message, length: 1 }
this	<window Obj>
message	"Hello, World!"

Current Operation: **Return Statement**

// checking var `name` in local memory

## Global Execution Context

EXECUTION PHASE

```
var name = "Luigi";  
let input = "Hello, World!";  
  
function broadcast(message) {  
  return `${name} says ${message}`.  
}
```

Global Memory	
Identifier	Value
window	<Global Object>
this	<window Object>

→ console.log(broadcast(input));

## Local Execution Context

EXECUTION PHASE

→ function broadcast(message) {  
 return `\${name} says \${message}`;  
}

Local Memory	
Identifier	Value
arguments	{ 0: message, length: 1 }
this	<window Obj>
message	"Hello, World!"

Current Operation: **Return Statement**

// `name` doesn't exist in local memory



## Global Execution Context

EXECUTION PHASE

```
var name = "Luigi";  
let input = "Hello, World!";  
  
function broadcast(message) {  
  return `${name} says ${message}`;  
}  
  
→ console.log(broadcast(input));
```

Global Memory	
Identifier	Value
window	<Global Object>
this	<window Object>
name	"Luigi"
input	"Hello, World!"
<b>broadcast</b>	<fn object>

EXECUTION PHASE

Local Memory	
Identifier	Value
arguments	{ message, length: 1 }
this	<window Obj>
message	"Hello, World!"

Current Operation: **Return Statement**

// look for var `name` in parent global memory

## Global Execution Context

EXECUTION PHASE

```
var name = "Luigi";  
let input = "Hello, World!";  
  
function broadcast(message) {  
  return `${name} says ${message}`;  
}  
  
→ console.log(broadcast(input));
```

Global Memory	
Identifier	Value
window	<Global Object>
this	<window Object>
name	"Luigi"
input	"Hello, World!"
broadcast	<fn object>

EXECUTION PHASE

Local Memory	
Identifier	Value
arguments	{ message, length: 1 }
this	<window Obj>
message	"Hello, World!"

Current Operation: **Return Statement**

// `name` exist in parent global memory

## Global Execution Context

EXECUTION PHASE

```
var name = "Luigi";  
let input = "Hello, World!";  
  
function broadcast(message) {  
  return `${name} says ${message}`.  
}
```

Global Memory	
Identifier	Value
window	<Global Object>
this	<window Object>

→ console.log(broadcast(input));

## Local Execution Context

EXECUTION PHASE

→ function broadcast(message) {  
 return `\${name} says \${message}`;  
}

Local Memory	
Identifier	Value
arguments	{ 0: message, length: 1 }
this	<window Obj>
message	"Hello, World!"

Current Operation: **Return Statement**

// takes value of `name` in parent global memory

## Global Execution Context

EXECUTION PHASE

```
var name = "Luigi";  
let input = "Hello, World!";  
  
function broadcast(message) {  
  return `${name} says ${message}`;  
}
```

Global Memory	
Identifier	Value
window	<Global Object>
this	<window Object>

→ console.log(broadcast(input));

## Local Execution Context

EXECUTION PHASE

→ function broadcast(message) {  
 return `Luigi says \${message}`;  
}

Local Memory	
Identifier	Value
arguments	{ 0: message, length: 1 }
this	<window Obj>
message	"Hello, World!"

Current Operation: **Return Statement**

// assign "Luigi" literal in \${name}

## Global Execution Context

EXECUTION PHASE

```
var name = "Luigi";  
let input = "Hello, World!";  
  
function broadcast(message) {  
  return `${name} says ${message}`;  
}
```

Global Memory	
Identifier	Value
window	<Global Object>
this	<window Object>

→ console.log(broadcast(input));

## Local Execution Context

EXECUTION PHASE

→ function broadcast(message) {  
 return `Luigi says \${message}`;  
}

Local Memory	
Identifier	Value
arguments	{ 0: message, length: 1 }
this	<window Obj>
message	"Hello, World!"

Current Operation: **Return Statement**

// checking var `message` in local memory

## Global Execution Context

EXECUTION PHASE

```
var name = "Luigi";  
let input = "Hello, World!";  
  
function broadcast(message) {  
  return `${name} says ${message}`;  
}
```

Global Memory	
Identifier	Value
window	<Global Object>
this	<window Object>

→ console.log(broadcast(input));

## Local Execution Context

EXECUTION PHASE

→ function broadcast(message) {  
 return `Luigi says \${message}`;  
}

Local Memory	
Identifier	Value
arguments	{ 0: message, length: 1 }
this	<window Obj>
message	"Hello, World!"

Current Operation: **Return Statement**

// `message` exist in local memory

## Global Execution Context

## EXECUTION PHASE

```
var name = "Luigi";  
let input = "Hello, World!";  
  
function broadcast(message) {  
  return `${name} says ${message}`;  
}
```

Global Memory	
Identifier	Value
window	<Global Object>
this	<window Object>

→ console.log(broadcast(input));

## Local Execution Context

## EXECUTION PHASE

→ function broadcast(message) {  
 return `Luigi says \${message}`;  
}

Local Memory	
Identifier	Value
arguments	{ 0: message, length: 1 }
this	<window Obj>
message	"Hello, World!"

Current Operation: **Return Statement**

// takes value of `message` in local memory

## Global Execution Context

## EXECUTION PHASE

```
var name = "Luigi";  
let input = "Hello, World!";  
  
function broadcast(message) {  
  return `${name} says ${message}`.  
}
```

Global Memory	
Identifier	Value
window	<Global Object>
this	<window Object>

→ console.log(broadcast(input));

## Local Execution Context

## EXECUTION PHASE

→ function broadcast(message) {  
 return "Luigi says Hello, World!";  
}

Local Memory	
Identifier	Value
arguments	{ 0: message, length: 1 }
this	<window Obj>
message	"Hello, World!"

Current Operation: **Return Statement**

// assign the value of `message` in `message`



## Global Execution Context

EXECUTION PHASE

```
var name = "Luigi";  
let input = "Hello, World!";  
  
function broadcast(message) {  
  return `${name} says ${message}`;  
}
```

Global Memory	
Identifier	Value
window	<Global Object>
this	<window Object>

→ console.log(broadcast(input));

## Local Execution Context

EXECUTION PHASE

```
function broadcast(message) {  
  return `${name} says ${message}`;  
}
```

Local Memory	
Identifier	Value
arguments	{ 0: message, length: 1 }
this	<window Obj>
message	"Hello, World!"

Current Operation: **Exit Function**

// return to parent execution context with value

→ // exit "Luigi says Hello, World!"

## Global Execution Context

EXECUTION PHASE

```
var name = "Luigi";  
let input = "Hello, World!";  
  
function broadcast(message) {  
  return `${name} says ${message}`;  
}  
  
console.log("Luigi says Hello, World!");
```

Global Memory	
Identifier	Value
window	<Global Object>
this	<window Object>
name	"Luigi"
input	"Hello, World!"
<b>broadcast</b>	<fn object>

EXECUTION PHASE

Local Memory	
Identifier	Value
arguments	{ message, length: 1 }
this	<window Obj>
message	"Hello, World!"

Current Operation: **Console Log Message**

```
// display `Luigi says Hello, World!` in console
```