


PRAVEEN  
000000

 x00

WORLD  
1-1

TIME  
30MIN

15+  
JS TRICKS.

2020 VUEHYD

 1 PLAYER GAME

2 PLAYER GAME

TOP- 000000



# WORLD 1 – BROWSER THINGS





# \$ - \$\$ - \$0

```
// first element matching .card
```

```
$(".card");
```

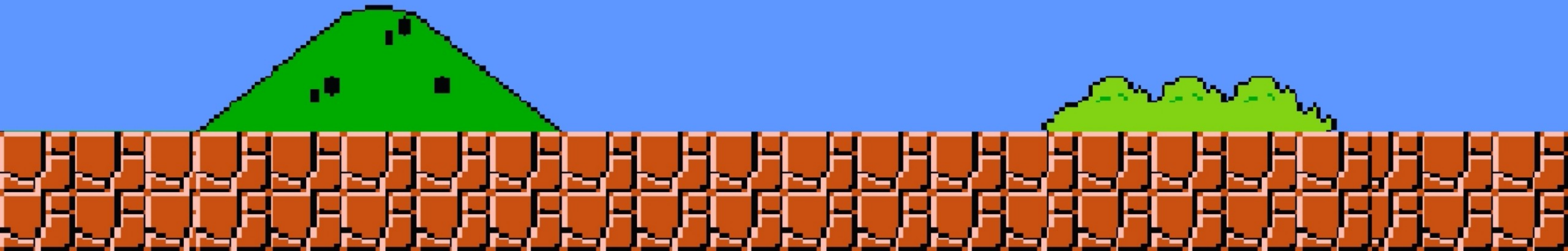
```
// all elements matching .card as an array
```

```
// and not NodeList
```

```
$$(".card");
```

```
// Currently inspected element
```

```
$0;
```

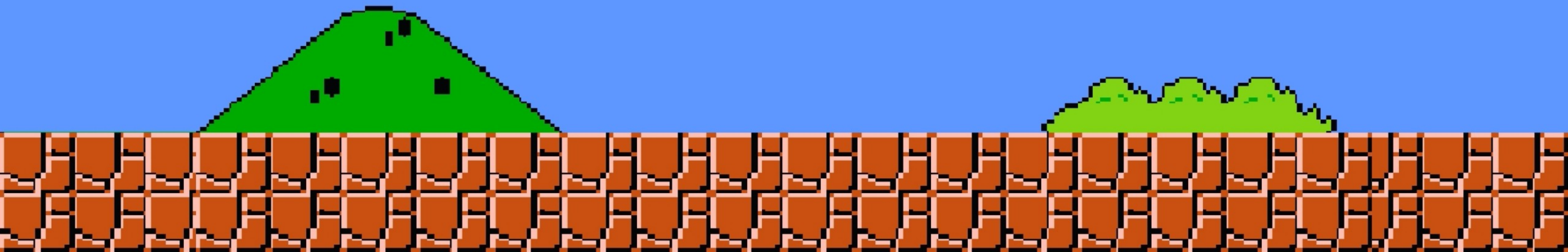


# CONSOLE STUFF

```
// See your records like... records
const users = [
  { name: "Praveen", loves: "Vue" },
  { name: "Aravind", loves: "React" }
];
console.table(users);

// Style your logs
console.log("%c WOW", "color: red; font-size: 100px");

// What did you even log?
const user = { name: "Praveen" };
const preferences = { zoom: 1.5 };
console.log({ user, preferences });
```





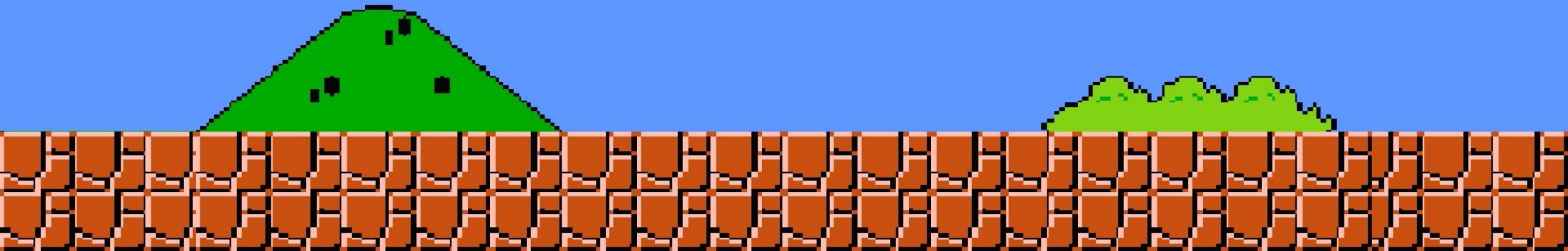
# COPY - INSPECT - EN/DECODE

```
// Get stuff right into your clipboard
const response = [
  { name: "Praveen", loves: "Vue" },
  { name: "Aravind", loves: "React" }
];
copy(response);

// Programmatically trigger inspection
inspect(document.body);

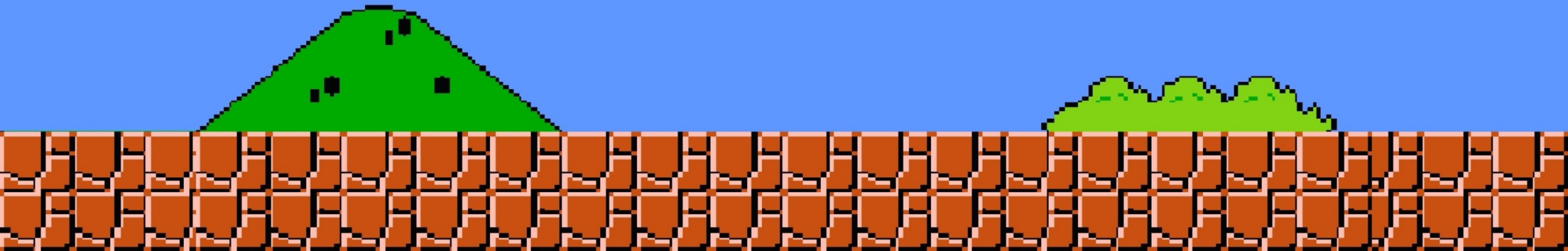
// Quickly get some encoded strings.
const encoded = btoa("Praveen Puglia");
const decoded = atob(encoded);

// Get a url to your favourite icon
const svg = $("svg.home").outerHTML;
const encodedIconURL = `data:image/svg+xml,${svg}`;
```



# DID YOU CLICK ME OR MY CHILDREN?

```
// Close a drawer when clicked anywhere outside.  
const drawer = document.querySelector(".drawer");  
document.body.addEventListener("click", e => {  
  if (drawer.contains(e.target)) {  
    return;  
  }  
  drawer.classList.remove("is-open");  
});
```

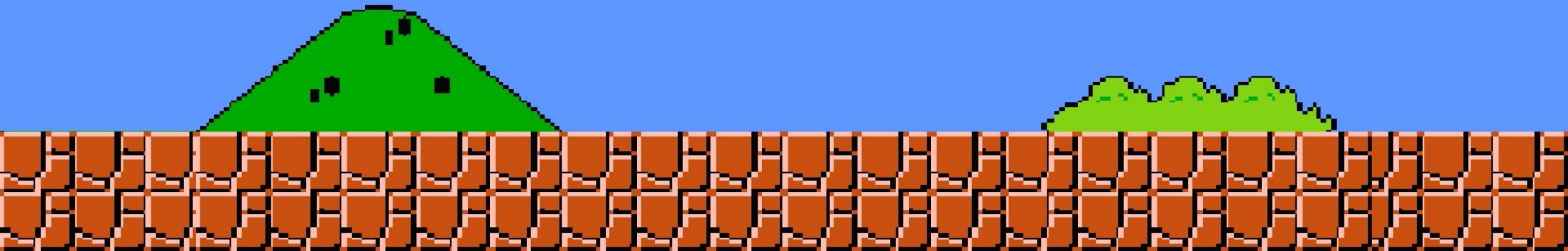




# GETTING QUERY PARAMS EASILY

```
// Assumed URL "https://praveenpuglia.com?post=121&action=edit"
const urlParams = new URLSearchParams(window.location.search);

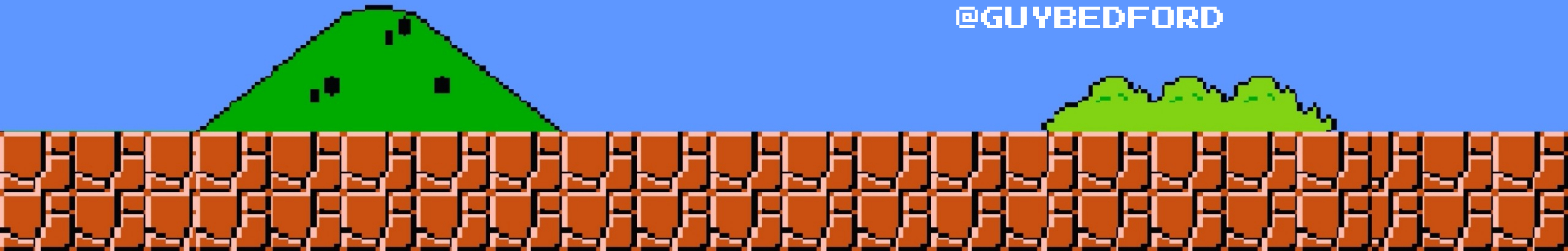
console.log(urlParams.has("post"));
// true
console.log(urlParams.get("action"));
// "edit"
console.log(urlParams.getAll("action"));
// ["edit"]
console.log(urlParams.toString());
// "?post=121&action=edit"
console.log(urlParams.append("active", "1"));
// "?post=121&action=edit&active=1"
```



# FROM NPM TO BROWSER

```
// Quickly add jQuery from npm for use  
// in console.  
await import("//dev.jspm.io/jquery");  
  
jQuery(".card");
```

@GUYBEDFORD





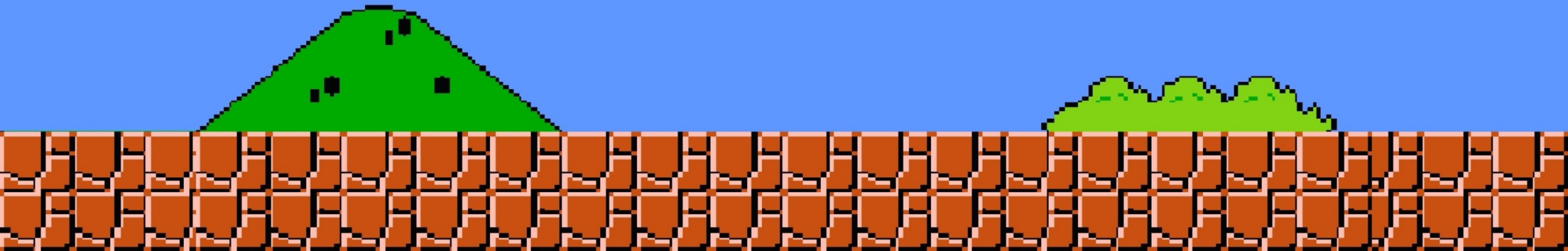


# WORLD 2 - JSON



# CHEAP COPY

```
const response = [  
  { name: "Praveen", loves: "Vue" },  
  { name: "Aravind", loves: "React" }  
];  
  
const clonedResponse =  
  JSON.parse(JSON.stringify(response));
```

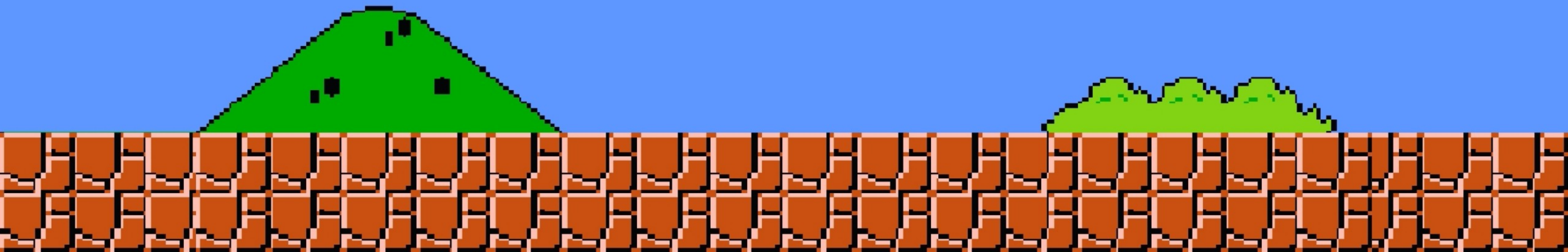




# PRETTIFY

```
const response = [{ name: "Praveen", loves: "Vue" }, { name: "Aravind", loves: "React" }];

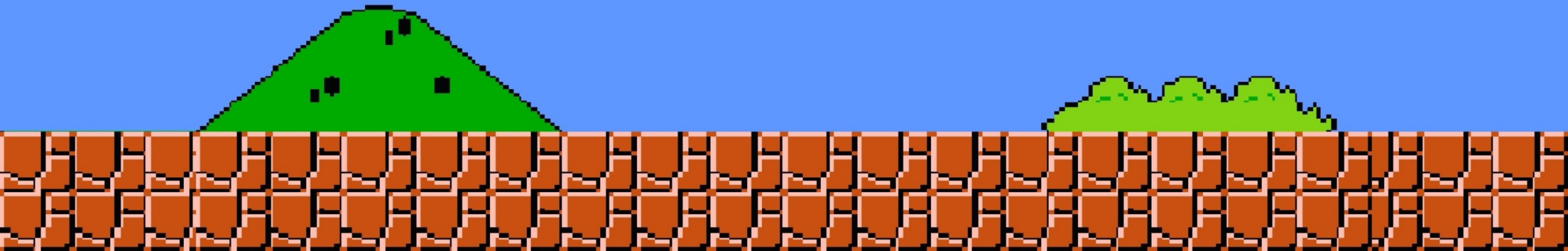
// Indent the json using 2 spaces
const prettified = JSON.stringify(response, null, 2);
// [
//   { name: "Praveen", loves: "Vue" },
//   { name: "Aravind", loves: "React" }
// ];
```



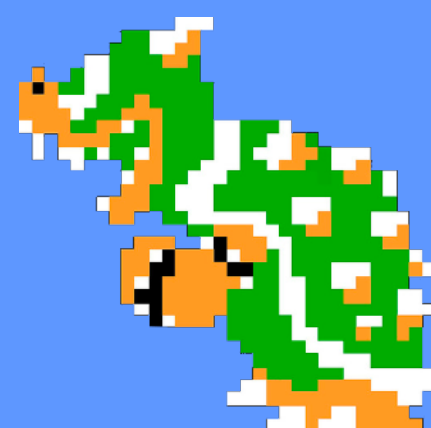
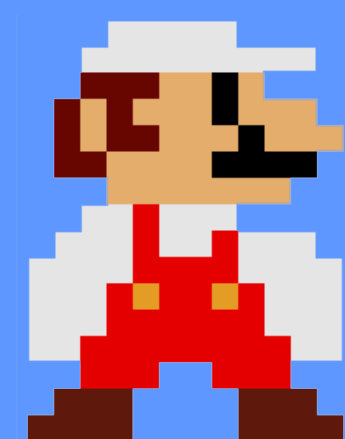
# FILTER PROPERTIES

```
const user = {
  firstName: "Praveen",
  lastName: "Puglia",
  age: 28,
  city: "Hyderabad"
};

const trimmedUser = JSON.parse(
  JSON.stringify(user, ["firstName", "age"], 2)
);
trimmedUser;
// {
//   firstName: "Praveen",
//   age: 28
// }
```





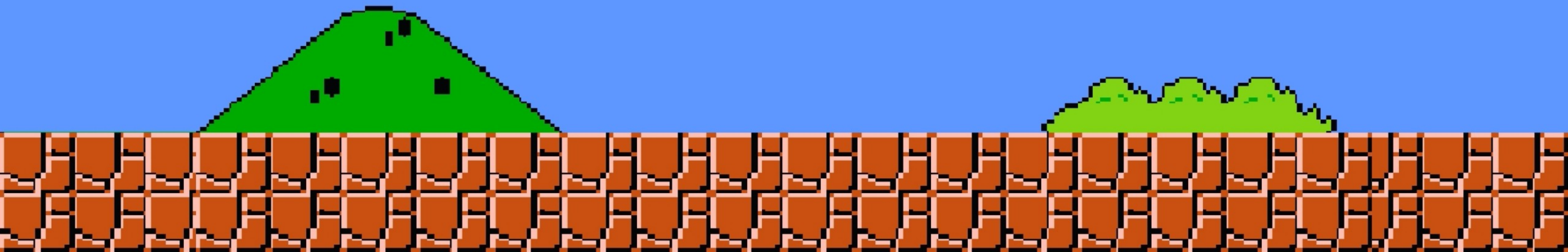


# WORLD 3 - LANGUAGE



# SWAP VARIABLES

```
let max = 3;  
let pivot = 5;  
if (max < pivot) {  
  // Swap Swap!  
  [max, pivot] = [pivot, max];  
}
```

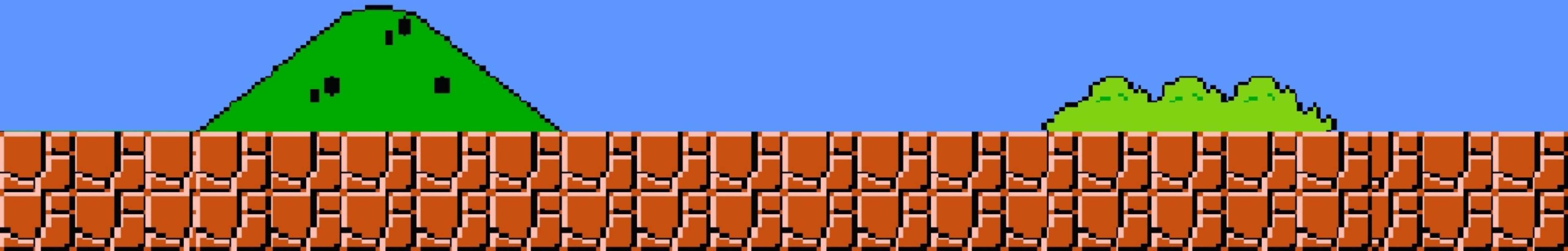




# PURE OBJECTS

```
const map = {};  
// Default object  
map.blahBlah();
```

```
const pureMap = Object.create(null);  
pureMap.onlyOwnProperty;
```

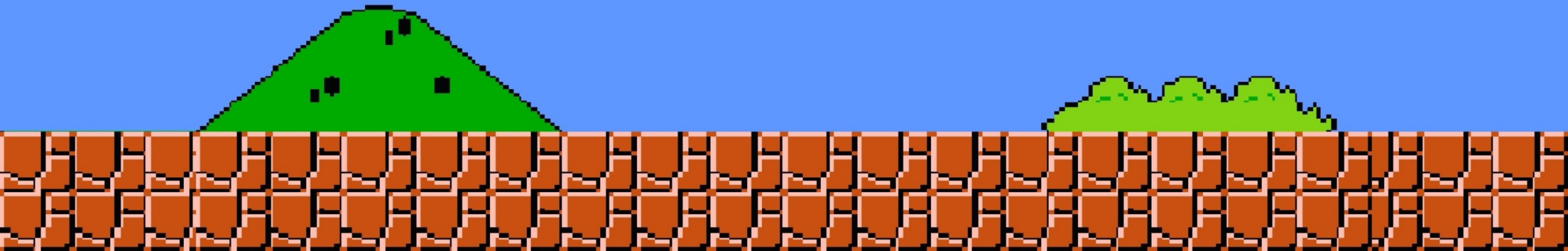


# REPEAT STRINGS

```
const dot = ".";
// Iterative way
let str = "";
for (let i = 0; i < 20; i++) {
  str += dot;
}

// Slightly better.
str = Array(20)
  .fill(dot)
  .join("");

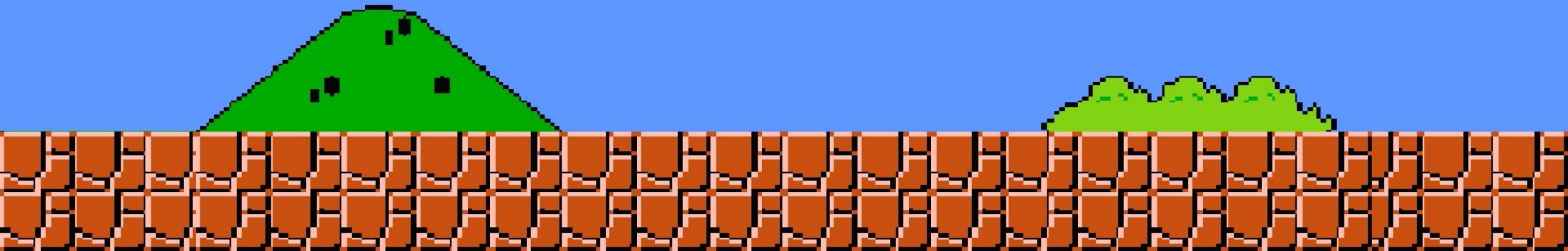
// Best
str = dot.repeat(20);
```





# GET LAST N ELEMENTS OF ARRAY

```
const arr = [1, 2, 3, 4, 5, 6];  
// Worst  
let last = [];  
const n = 3;  
for (let i = arr.length - n; i < arr.length; i++) {  
  last.push(arr[i]);  
}  
  
// One-liner  
last = arr.slice(-3);
```



# GET UNIQUE VALUES FROM AN ARRAY

```
const arr = [1, 2, 3, 1, 2, 3, 12, 5, 7, 8, 5];
```

```
// How beginners do it.
```

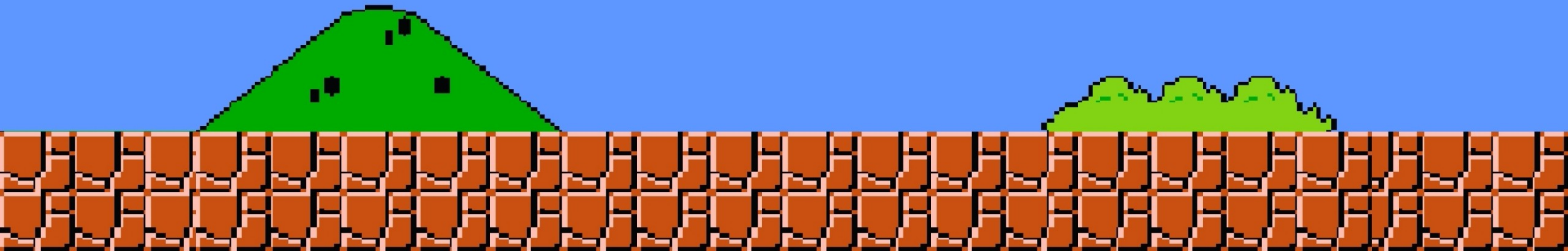
```
let uniq = [];
```

```
arr.forEach(item => {  
  !uniq.includes(item) && uniq.push(item);  
});
```

```
// How veterans do it.
```

```
uniq = [...new Set(arr)];
```

```
uniq; // [1, 2, 3, 12, 5, 7, 8]
```





# DELETE LAST n ELEMENTS FROM ARRAY

```
const arr = [1, 2, 3, 1, 2, 3, 12, 5, 7, 8, 5];
```

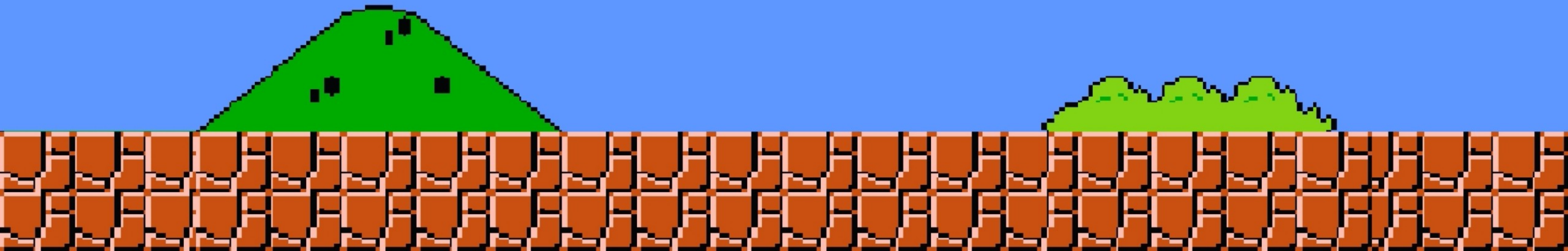
```
// The long road.
```

```
const n = 3;
```

```
for (let i = 0; i < n; i++) {  
  arr.pop();  
}
```

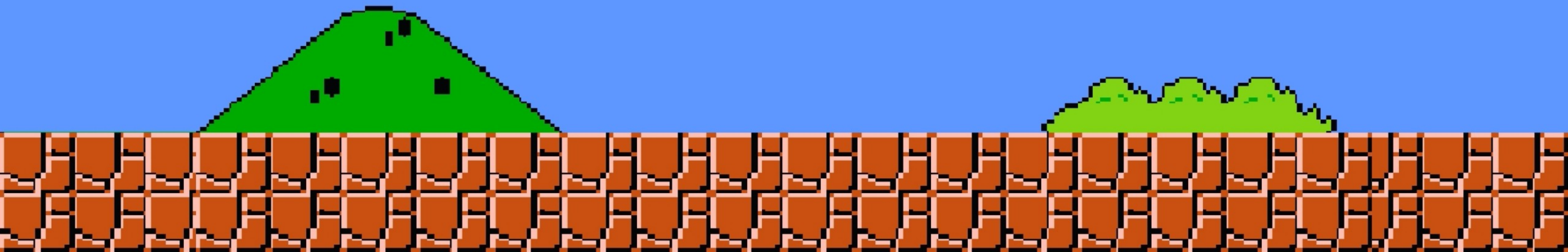
```
// The gully!
```

```
arr.length = arr.length - n;
```



# FALSY VALUES IN ARRAYS ?

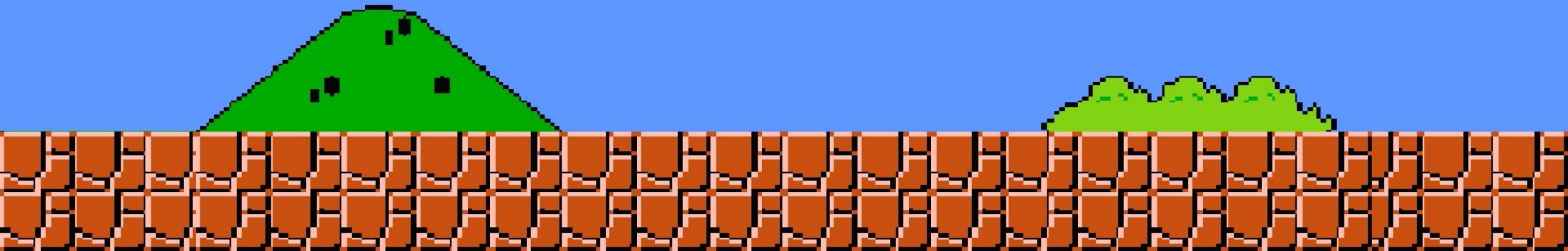
```
let vals = [true, false, 0, "", undefined, 1, NaN, 2];  
  
// Remove all falsy values  
vals = vals.filter(Boolean);  
  
// Keep zeros!  
vals = vals.filter(e => e === 0 || e);
```





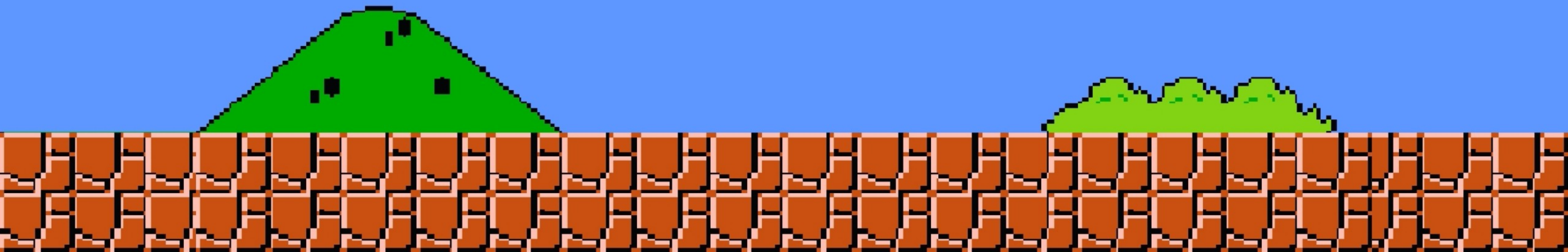
# REQUIRED PARAMETERS

```
function render(el, container) {  
  if (!el) {  
    throw new Error("el is required");  
  }  
  if (!container) {  
    throw new Error("container is required");  
  }  
  // Real code starts here.  
}  
  
const required = () => {  
  throw new Error("missing required params");  
};  
function render(el = required(), container = required()) {  
  // Real code right away!  
}
```



# TAGGED TEMPLATE LITERALS

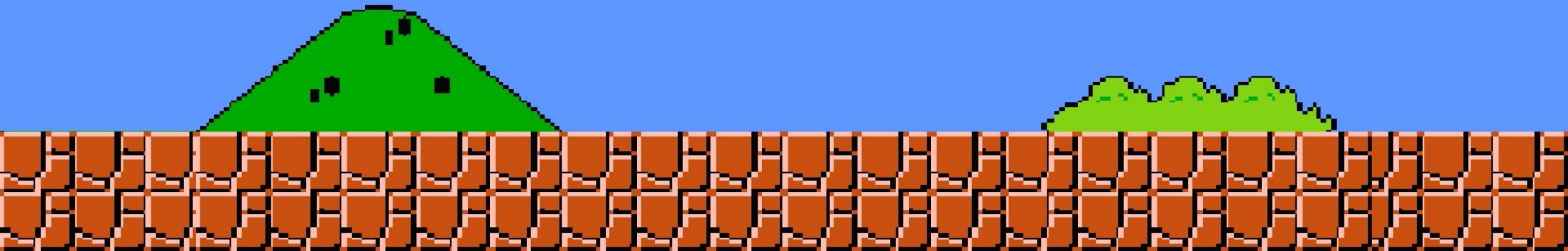
```
const styled = {  
  button(styles, ...params) {  
    const css = styles[0].replace(/\n/g, "");  
    return `  }  
};  
  
const button = styled.button`  
  background: rebeccapurple;  
  font-size: 18px;  
  font-weight: bold;  
  border-radius: 1rem;  
  color: white;  
  padding: 0.5rem 1rem;  
`;  
  
document.body.innerHTML = button;
```





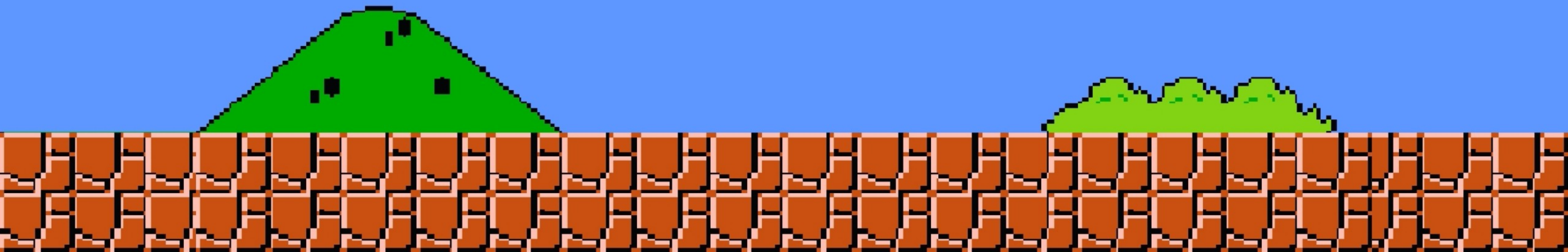
# NULLish COALESCING

```
const res = {  
  data: null,  
  attempts: 0  
};  
// Oh my my! data is null!  
const data = res.data ?? {};  
const attempts = res.attempts ?? 10; // 0
```



# OPTIONAL CHAINING

```
const res = {  
  data: null,  
  attempts: 0  
};  
// Oh my my! data is null!  
const name = res.data?.name ?? "Unknown";  
// non existent methods  
res.nonExistent?.();
```





# TAKE A BREAK

@PRAVEENPUGLIA  
PRAVEENPUGLIA.COM

