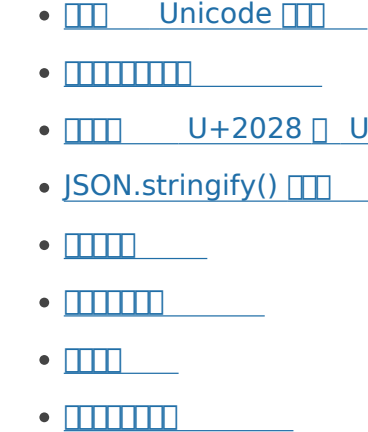




```
"\uD842\uDFB7"  
// "𐀢𐀦"  
  
"\u20BB"  
// " 7"
```

[illegible]

"\u{20BB7}"

// "00"

"\u{41}\u{42}\u{43}"

```
// "ABC"
```

```
let hello = 123;
```

```
hell\u{6F} // 123
```

```
'\u{1F680}' === '\uD83D\uDE80'
```

```
// true
```

UTF-16

--	--	--	--	--	--

[illegible]

```
'\z' == 'z' // true
```

```
'\172' == 'z' // true
```

```
'\x7A' == 'z' // true
```

```
'\u007A' == 'z' // true
```

```
'\u{7A}' == 'z' // true
```

--	--	--	--	--	--	--	--

[illegible]

```
Iterator [ ]
```

for...of ☐ ☐ ☐ ☐ ☐

```
for (let codePoint of 'foo') {
```

```
console.log(codePoint)
```

}

// "f"

// "0"

// "0"

[illegible]

0xFFFF 

for

```
let text = String.fromCodePoint(0x20BB7);
```

```
for (let i = 0; i < text.length; i++) {
```

```

    console.log(text[i]);
  }
  // " "
  // " "

  for (let i of text) {
    console.log(i);
  }
  // " "

```

□□□□□□□□

text □□□□□□□□

for □□□□□□□□□□□□□□□□

for...of

□□□□□□□□□□

□□□□

U+2028 □ U+2029

JavaScript □□□□□□□□□□□□□□□□□□□□
 U+4e2d□□□□□□□□□□□□□□□□□□□□

“ ” Unicode □□
 \u4e2d □□□□□□

```
' ' === '\u4e2d' // true
```

□□ JavaScript □□ 5□□□□□□□□□□□□□□□□□□□□

- U+005C□□□□ reverse solidus)
- U+000D□□□□ carriage return□
- U+2028□□□□□ line separator□
- U+2029□□□□□ paragraph separator□
- U+000A□□□□□ line feed□

□□□□□□□□□□□□□□□□□□□□□□□□

\\ □□

\u005c □

□□□□□□□□□□

JSON □□□□□□□□□□

U+2028□□□□□□

U+2029

□□□□□□□□□□

JSON □ JSON.parse □□□□□□□□

```

const json = '"\u2028"';
JSON.parse(json); // □□□□

```

JSON □□□□□ RFC 7159□□□□□□□□□□□□□□

[ES2019](#) □ JavaScript

□□□□□

U+2028□□□□□□

U+2029□□□□□□

```
const PS = eval('"'\u2029'');
```



```
$('#result').append(`
  There are <b>${basket.count}</b> items
  in your basket, <em>${basket.onSale}</em>
  are on sale!
`);
```

template string`

```
// 
`In JavaScript '\n' is a line-feed.`

// 
`In JavaScript this is
not legal.`

console.log(`string text line 1
string text line 2`);

// 
let name = "Bob", time = "today";
`Hello ${name}, how are you ${time}?`
```

```
let greeting = ``Yo` World!`;
```

```
$('#list').html(`
<ul>
  <li>first</li>
  <li>second</li>
</ul>
`);
```


trim

```
$('#list').html(`
<ul>
  <li>first</li>
  <li>second</li>
</ul>
`.trim());
```

```
function authorize(user, action) {
  if (!user.hasPrivilege(action)) {
    throw new Error(
      // 权限不足
      // 'User '
      // + user.name
      // + ' is not authorized to do '
      // + action
      // + '.'
      `User ${user.name} is not authorized to do ${action}.`);
  }
}
```

```
let x = 1;

let y = 2;

`${x} + ${y} = ${x + y}`

// "1 + 2 = 3"

`${x} + ${y * 2} = ${x + y * 2}`

// "1 + 4 = 5"

let obj = {x: 1, y: 2};

`${obj.x + obj.y}`

// "3"
```

```
function fn() {  
  return "Hello World";  
}
```

```
}

`foo ${fn()} bar`
// foo Hello World bar
```

```
████████████████████████████████████████████████████████████████████████████████
[] toString []
```

```
████████████████████████████████████████
```

```
// []place[]
let msg = `Hello, ${place}`;
// []
```

```
████████████████████████████████████████████████████████████████████████████████
████████████████████████████████████████████████████████████████████████████████
```

JavaScript

```
`Hello ${'World'}`
// "Hello World"
```

```
████████████████████████████████████████
```

```
const tpl = addr => `
  <table>
    ${addr.map(addr => `
      <tr><td>${addr.first}</td></tr>
      <tr><td>${addr.last}</td></tr>
    `).join('')}
  </table>
`;
```

```
████████████████████████████████████████████████████████████████████████████████
```

```
const data = [
  { first: '<Jane>', last: 'Bond' },
  { first: 'Lars', last: '<Croft>' },
];

console.log(tpl(data));
// <table>
//
//   <tr><td><Jane></td></tr>
```

```
//      <tr><td>Bond</td></tr>
//
//      <tr><td>Lars</td></tr>
//      <tr><td><Croft></td></tr>
//
// </table>
```

```
let func = (name) => `Hello ${name}!`;
func('Jack') // "Hello Jack!"
```

[illegible]

--	--	--	--	--	--	--

```
let template = `
<ul>
  <% for(let i=0; i < data.supplies.length; i++) { %>
    <li><%= data.supplies[i] %></li>
  <% } %>
</ul>
`;
```

%> ☐ JavaScript ☐☐☐

<%...%> ☐ JavaScript ☐ ☐ ☐ ☐

 $\leq \dots$

JavaScript

```
echo('<ul>');

for(let i=0; i < data.supplies.length; i++) {
  echo('<li>');
  echo(data.supplies[i]);
  echo('</li>');
};

echo('</ul>');
```

```
let evalExpr = /<%= (.+?) %>/g;
let expr = /<%([\\s\\S]+?) %>/g;

template = template
    .replace(evalExpr, ``); \n echo( $1 ); \n echo(``)
    .replace(expr, ``); \n $1 \n echo(``);

template = 'echo(`` + template + ``)';
```

```
let script =
  `(function parse(data){
    let output = "";

    function echo(html){
      output += html;
    }

    ${ template }

    return output;
  })`;

return script;
```

```
function compile(template){
  const evalExpr = /<%= (.+?) %>/g;
  const expr = /<%([\\s\\S]+?) %>/g;

  template = template
    .replace(evalExpr, '`'); \n echo( $1 ); \n echo(``)
    .replace(expr, '`'); \n $1 \n echo(``);

  template = 'echo(`` + template + ``)';

  let script =
```

```
`(function parse(data){
  let output = "";

  function echo(html){
    output += html;
  }

  ${ template }

  return output;
})`;

return script;
}
```

compile ☐☐☐☐☐☐☐☐

```
let parse = eval(compile(template));
div.innerHTML = parse({ supplies: [ "broom", "mop", "cleaner" ] });
//   <ul>
//     <li>broom</li>
//     <li>mop</li>
//     <li>cleaner</li>
//   </ul>
```



☐☐☐☐ " ☐☐☐ " ☐☐ tagged template ☐☐

```
alert`hello`
//  ☐☐☐
alert(['hello'])
```



" ☐☐ "



```
let a = 5;
let b = 10;

tag`Hello ${ a + b } world ${ a * b }`;

// 
tag(['Hello ', ' world ', ''], 15, 50);
```

tag

tag

tag

```
function tag(stringArr, value1, value2){
  // ...
}

// 

function tag(stringArr, ...values){
  // ...
}
```

tag

tag

tag value1 value2

tag

- ['Hello ', ' world ', '']
- : 15
- 50

tag

```
tag(['Hello ', ' world ', ''], 15, 50)
```

tag

tag

```
let a = 5;
let b = 10;
```


passthru rest

```
function passthru(literals, ...values) {
  let output = "";
  let index;
  for (index = 0; index < values.length; index++) {
    output += literals[index] + values[index];
  }

  output += literals[index]
  return output;
}
```

""
"XXXXXXXXXXXX"
HTML

```
let message =
  SaferHTML`<p>${sender} has sent you a message.</p>`;

function SaferHTML(templateData) {
  let s = templateData[0];
  for (let i = 1; i < arguments.length; i++) {
    let arg = String(arguments[i]);

    // Escape special characters in the substitution.
    s += arg.replace(/&/g, "&amp;")
              .replace(/</g, "&lt;")
              .replace(/>/g, "&gt;");

    // Don't escape special characters in the template.
    s += templateData[i];
  }
  return s;
}
```

sender SaferHTML

```
let sender = '<script>alert("abc")</script>'; // 
let message = SaferHTML`<p>${sender} has sent you a message.</p>`;

message
// <p>&lt;script&gt;alert("abc")&lt;/script&gt; has sent you a message.</p>
```

```
118n`Welcome to ${siteName}, you are visitor number ${visitorNumber}!\n\n// "xxxxxxx"

```

Mustache

```
// 编译hashTemplate
// 编译模板

let libraryHtml = hashTemplate`

  <ul>

    #for book in ${myBooks}

      <li><i>#{book.title}</i> by #{book.author}</li>

    #end

  </ul>

`;
```

JavaScript

```
jsx`
<div>
  <input
    ref='input'
    onChange='${this.handleChange}'
    defaultValue='${this.state.value}' />
    ${this.state.value}
  </div>
`
```

--	--	--	--	--	--	--

jsx ☐ ☐ ☐ ☐ ☐ ☐

DOM

--	--	--	--	--

React ☐ ☐ ☐ ☐ ☐ ☐

GitHub

jsx

444

1

java

JavaScript

Java 

```
java`  
class HelloWorldApp {  
    public static void main(String[] args) {  
        System.out.println("Hello World!"); // Display the string.  
    }  
}
```

```
HelloWorldApp.main();
```

```
console.log`123`  
// ["123", raw: Array[1]]
```

```
tag`First line\nSecond line`

function tag(strings) {
  console.log(strings.raw[0]);
  // strings.raw[0] = "First line\\nSecond line"
  // [] [] [] "First line\nSecond line"
}
```

--	--	--	--	--	--	--	--

```
function latex(strings) {
  // ...
}

let document = latex`
\newcommand{\fun}{\textbf{Fun!}} // 関数
\newcommand{\unicode}{\textbf{Unicode!}} // 文字
\newcommand{\xerxes}{\textbf{King!}} // 王
```

Breve over the h goes \u{h}ere //

,

document

LaTeX

JavaScript

\u00FF

\u{42}

Unicode

\unicode

\x56

\xerxes

\u

\x

LaTeX

JavaScript

ES2018

undefined

raw

```
function tag(strs) {  
  strs[0] === undefined  
  strs.raw[0] === "\\unicode and \\u{55}";  
}  
tag`\\unicode and \\u{55}`
```

JavaScript

undefined

raw

tag

```
let bad = `bad escape sequence: \unicode`; // 
```

#2

2020-04-16 10:29:52 UTC

2021-10-19 09:07:31 UTC