Paver Picker API

Viewer attributes

Attribute	Value	Description
`data-pp-viewer` *(Required)*	Product code or product ID	Generates viewer html for the requested product within the parent element.
`data-pp-controlsid`	A chosen unique ID e.g. `controls1`	Specifies which set of controls the viewer is bound to.
`data-pp-scale`	`0.125`, `0.25`, `0.5`, `1` or `2`	Sets the initial scale of the viewer. Uses the products default scale if not set.
`data-pp-scene`	Scene name (Available at https://www.paverpicker.com/index.php/scenes/)	Displays the product within the specified scene image.
`data-pp-hasfullscreen`	*(No value)*	Adds a full-screen icon button to the viewer.
`data-pp-haszoom`	*(No value)*	Adds a zoom-in and zoom-out icon buttons to the viewer.

Controls attributes

Attribute	Value	Description
`data-pp-controlsid` *(Required)*	A chosen unique ID e.g. `controls1`	Specifies which viewer the controls are bound to.

`data-pp-controls` *(Required)*	A list comprising of at least one of the following, each separated by a comma:
	`layout`
	`layoutVariation`
	`angle`
	`jointColour`
	`jointWidth`
	e.g. `layout,angle,jointWidth`
		Specifies which controls are to be displayed.

Swatch attributes

Attribute	Value	Description
`data-pp-swatch` *(Required)*	Product code or product ID	Generates swatch html for the requested product within the parent element.
`data-pp-width` *(Required)*	Desired width value (Max. 500)	Sets the width of the swatch.
`data-pp-height` *(Required)*	Desired height value (Max. 500)	Sets the height of the swatch.
`data-pp-scale`	`0.125`, `0.25`, `0.5`, `1` or `2`	Sets the scale of the swatch. Uses the products default scale if not set.

Viewer JavaScript Functions

Function	Description
`PaverPicker.load()`	Use when viewers are loaded dynamically, call the function after the dynamic event.
`PaverPicker.setProduct(viewerID, sceneID, animate)`	Used to change a product.
`PaverPicker.update(viewerID, animate)`	Updates the specified viewer. Particularly useful for updating the scene in the viewer. It may also need to be used with responsive websites on window 'resize' and 'orientationchange'.

parameters

`viewerID` Type: String

`sceneID` Type: String

`Animate` Type: Boolean

What is it?

Paver Picker website integration is a way to combine the accessibility of your website with the powerful, interactive Paver Picker product visualization tools. The benefits of integrating the Paver Picker software into your website are:

- Full Control over analytics
- Display of quality high resolution product and scene images
- Your technical documents can be accessed beside the product
- Fully customisable to your website’s layout and styling
- No need to open Paver Picker software in a new tab or pop-up

How does it work?

Set-up is simple. We show you a quick set-up guide and thereafter provide you with some tips to get the best appearance for your products. The generated elements will have default CSS styling but can be overridden to match the websites appearance.

Let's get started…

Integration Example

1. Linking to the script file

The first step to getting the Paver Picker API running within a webpage is to enter the following script tag before the closing `</body>` tag:

HTML

```html
<script
src="https://www.paverpicker.com/paverpicker3/dist/bundle.min.js?username=YourUsername"
></script>
```

This allows the Paver Picker JavaScript file to run within your webpage.

2. Creating a basic viewer

The next step is to insert a small snippet of code where you would like the viewer to display on your webpage:

HTML

```html
<div data-pp-viewer="Code/ID"></div>
```

The JavaScript will automatically scan for the `data-pp-viewer` value to know what product to display.

You will also need to give the element a width and height:

CSS

```css
[data-pp-viewer] {
    width: 400px;
    height: 300px;
}
```

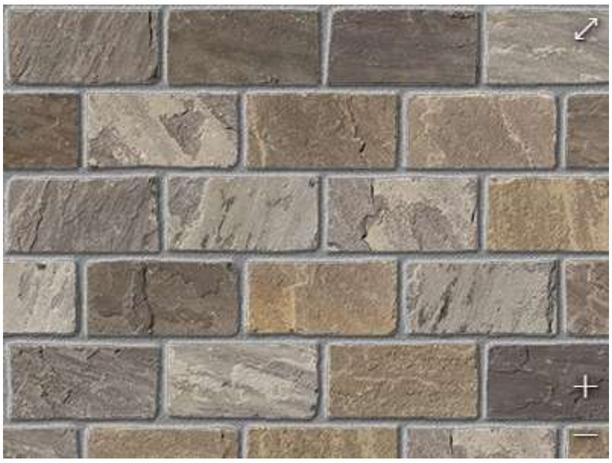
Your page should now display a viewer like this:

3. Viewer display options and controls

Next, let's change the scale of the viewer and add zoom and full-screen controls:

```HTML
<div data-pp-viewer="Code/ID" data-pp-scale="0.5" data-pp-hasfullscreen data-pp-haszoom
></div>
```

It should now display like this:

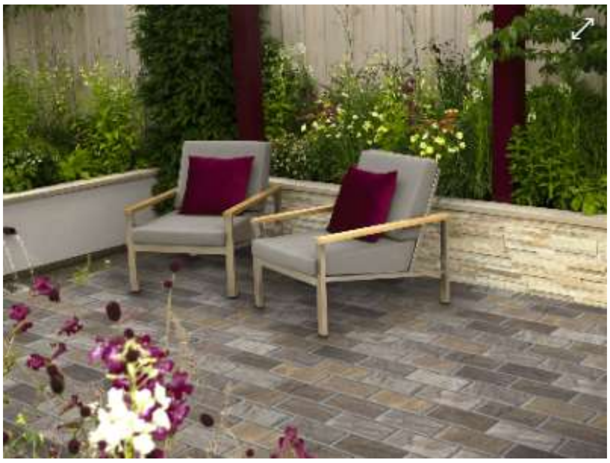


4. Setting a scene

Optionally, you can set a scene image to display the product in (note that if the `data-pp-scale` and `data-pp-haszoom` attributes are set, they will be ignored when using a scene):

```HTML
<div data-pp-viewer="Code/ID" data-pp-hasfullscreen data-pp-scene="Patio 1" ></div>
```

It should display a scene like this:



5. Viewer product controls

To add controls that change the way the product is presented, insert the following code where you would like the controls to display on your webpage:

HTML

```html
<div data-pp-controls="layout,layoutVariation,angle,jointColour,jointWidth" data-pp-
controlsId="1"></div>
```

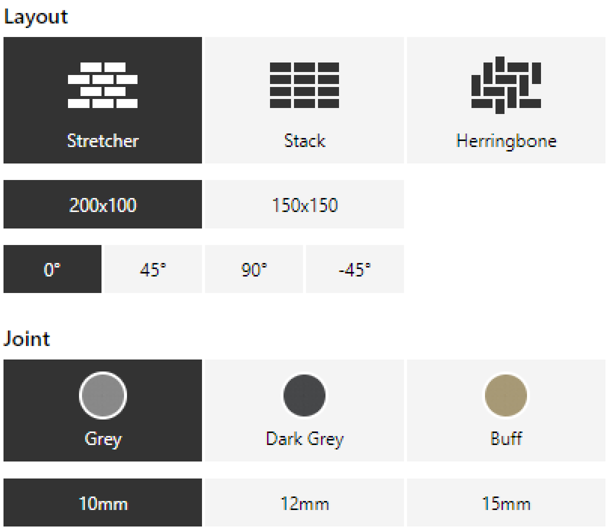
The `data-pp-controlsId` value can be any unique ID you choose.

You must also add a `data-pp-controlsId` attribute with the same value to the viewer element:

HTML

```html
<div data-pp-viewer="Code/ID" data-pp-controlsId"1" ></div>
```

You should now have controls like this:



6. Creating a swatch list

Another way to display a product is as a swatch, they are typically used for listing products. To add swatches, insert code like this where you would like each one to display:

HTML

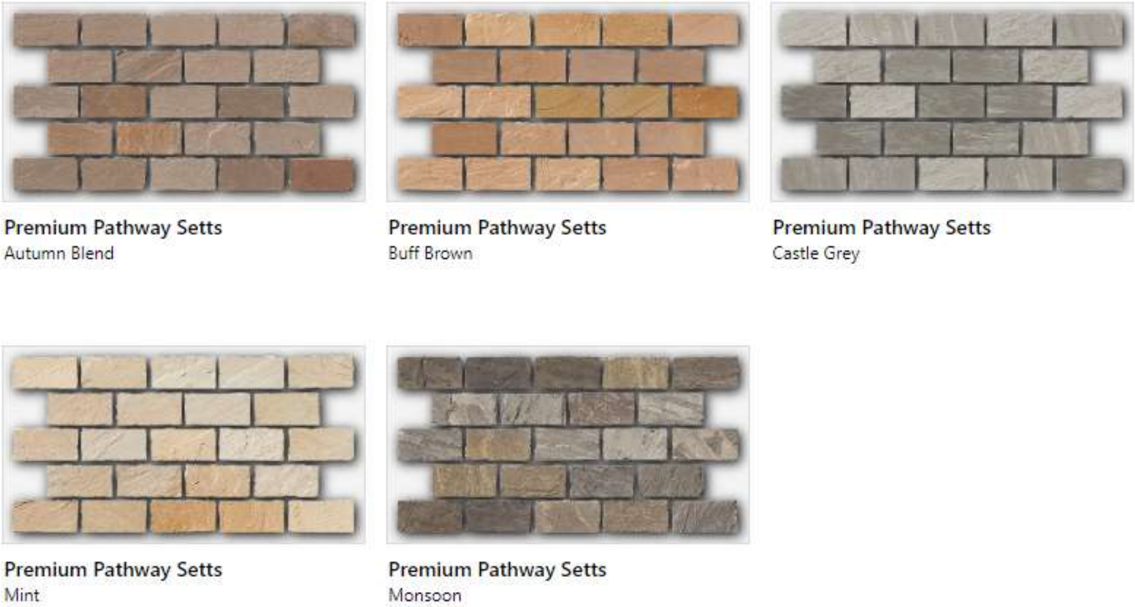
```html
<div data-pp-swatch="Code/ID" data-pp-scale="0.25" data-pp-width="263" data-pp-height="138"></div>
```

You can add styling to make the list display as desired:

CSS

```css
[data-pp-swatch] {
    float: left;
}
```

The list should look like this:



Tips

Accurately positioning products within a viewer

You can match the viewer size to the product size by using the following calculations:

Width = (product width + joint width) x columns x scale
Height = (product height + joint width) x rows x scale

If the product is 240mm wide and 120mm high, the joint is 10mm wide, displaying 3 columns and 5 rows with a scale of 0.5, the calculations are as follows:

Width = (200 + 10) x 4 x 0.5 = 420
Height = (100 + 10) x 5 x 0.5 = 275

These sizes can be set as an absolute value:

CSS

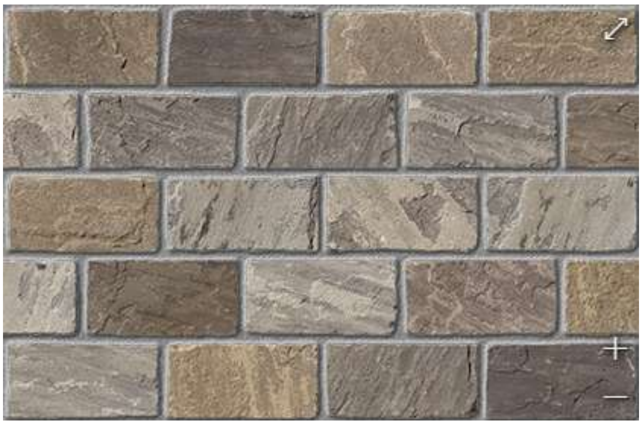
```css
div {
    width: 420px;
    height: 275px;
}
```

Or using the CSS `calc()` function:

CSS

```css
div {
    width: calc((200px + 10px) * 4 * 0.5)
    height: calc((100px + 10px) * 5 * 0.5)
}
```

With these values set, the viewer should look like this:



If scenes are to be viewed in the viewer then please also consider that the scenes are of ratio 3:2.

Responsive swatch scaling

For responsive websites you can add a small snippet of JavaScript to make swatches automatically scale to the parent containers width.

If for example the swatches were in a parent element with `id="list"` and the swatches have the property `data-pp-width="263"`, the JavaScript would be as follows:

JS

```js
window.addEventListener("resize", onResize);
window.addEventListener("orientationchange", onResize);
function onResize() {
    const list = document.getElementById("list");
    const swatchWidth = 263;
    const columnCount = Math.ceil(list.offsetWidth / swatchWidth);
    for (item of list.children) {
        item.style.width = 100 / columnCount + "%";
    }
}
onResize();
```