

build custom element

Polymer 1.x	Polymer({})
Polymer 2.0	class MyElement extends Polymer.Element {}

API

this.shadowRoot	get shadow DOM
this.\$.container	access identifier #container
<template is="-dom-bind">	use Polymer bindings without defining a new custom element
\$\$(<selector>)	Returns the first node in this element's local DOM that matches selector

instance methods <https://www.polymer-project.org/1.0/docs/devguide/instance-methods>

Instance Methods

fire(type, [detail], [options])	Fires a custom event
async(method, [wait])	Calls method asynchronously
debounce(jobName, callback, [wait])	Call debounce to collapse multiple requests for a named task into one invocation
toggleAttribute(name, bool, [node])	toggles the named boolean attribute
this.transform('rotateX(90deg)', this.\$.myDiv)	Applies a CSS transform to the specified node, or host element if no node is specified
resolveUrl(url)	returns a path relative to the current document

instance methods <https://www.polymer-project.org/1.0/docs/devguide/instance-methods>

property list for polymer base

<https://www.polymer-project.org/1.0/docs/api/Polymer.Base>

Polymer.Templatizer behavior

```
// Get a template from somewhere, e.g. light DOM
var template = Polymer.dom(this).querySelector('template');
// Prepare the template
this.templateize(template);
// Instance the template with an initial data model
var instance = this.stamp({ myProp: 'initial' });
// Insert the instance's DOM somewhere, e.g. light DOM
Polymer.dom(this).appendChild(instance.root);
// Changing a property on the instance will propagate to bindings
// in the template
instance.myProp = 'new value';
```

The Polymer.Templatizer behavior adds methods to generate instances of templates that are each managed by an anonymous Polymer.Base instance.

<https://www.polymer-project.org/1.0/docs/api/Polymer.Templatizer>

Lifecycle

created
ready
attached
detached
attributeChanged

Property name to attribute name mapping

Attribute names are converted to lowercase property names

Attribute names with dashes are converted to camelCase property names

Properties <https://www.polymer-project.org/1.0/docs/devguide/properties>



By **brian (brianyang)**
cheatography.com/brianyang/
brianyang.com

Not published yet.
Last updated 22nd September, 2016.
Page 1 of 3.

Sponsored by **CrosswordCheats.com**
Learn to solve cryptic crosswords!
<http://crosswordcheats.com>

Custom CSS properties

```
.title {
  color: var(--my-toolbar-title-color);
}
```

value of the property will inherit down to the toolbar where it is used if defined

<https://www.polymer-project.org/1.0/docs/devguide/styling>

Declared properties

computed The value is interpreted as a method name and argument list

observer The value is interpreted as a method name to be invoked when the property value changes

Configuring default property values

```
Polymer({
  is: 'x-cus tom',
  properties: {
    mode: {
      type: String,
      value: 'auto'
    },
    data: {
      type: Object,
      notify: true,
      value: function() { return {}; }
    }
  }
});
```

Style

:host ::content div must have a selector to the left of the ::content pseudo-element

--my-toolbar-title-color These custom properties can be defined and will propagate down the composed DOM tree

color: var(--my-toolbar-title-color, blue) include a default value in the var() function

Custom CSS mixins

```
// example
<dom-module id="my-toolbar">
  <template>
    <style>
      :host {
        padding: 4px;
        background-color: gray;
        apply a mixin /
        @apply(--my-toolbar-theme);
      }
      .title {
        @apply(--my-toolbar-title-theme);
      }
    </style>
    <span class="title">{{title}}</span>
  </template>
  ...
</dom-module>
// example usage
<dom-module id="my-element">
  <template>
    <style>
```



By **brian (brianyang)**

cheatography.com/brianyang/

brianyang.com

Not published yet.

Last updated 22nd September, 2016.

Page 2 of 3.

Sponsored by **CrosswordCheats.com**

Learn to solve cryptic crosswords!

<http://crosswordcheats.com>

Custom CSS mixins (cont)

```
/ Apply custom theme to toolbars /
:host {
  --my-toolbar-theme: {
    background-color: green;
    border-radius: 4px;
    border: 1px solid gray;
  };
  --my-toolbar-title-theme: {
    color: green;
  };
}

Make only toolbars with the .warning
class red and bold /
.warning {
  --my-toolbar-title-theme: {
    color: red;
    font-weight: bold;
  };
}

</style>
<my-toolbar title= "This one is
green." > </my-toolbar>
<my-toolbar title= "This one is green
too."></my-toolbar>
<my-toolbar class= " warning"
title= "This one is red."></my-toolbar>
</template>
<script>
  Polymer({ is: 'my-element' });
</script>
</demo>
```



By **brian (brianyang)**

cheatography.com/brianyang/
brianyang.com

Not published yet.

Last updated 22nd September, 2016.

Page 3 of 3.

Sponsored by **CrosswordCheats.com**

Learn to solve cryptic crosswords!

<http://crosswordcheats.com>