

## **pr**r Build Status

An sensible alternative to `Object.defineProperty()`.  
Available in npm and Ender as **pr**r.

### **Usage**

Set the property 'foo' (`obj.foo`) to have the value 'bar' with default options ('enumerable', 'configurable' and 'writable' are all false):

```
pr(r(obj, 'foo', 'bar'))
```

Adjust the default options:

```
pr(r(obj, 'foo', 'bar', { enumerable:  
    true, writable: true }))
```

Do the same operation for multiple properties:

```
pr(r(obj, { one: 'one', two: 'two' }))  
// or with options:  
pr(r(obj, { one: 'one', two: 'two' }, {  
    enumerable: true, writable:  
    true }))
```

## Simplify!

But obviously, having to write out the full options object makes it nearly as bad as the original `Object.defineProperty()` so we can simplify.

As an alternative method we can use an options string where each character represents a option: 'e'=='enumerable', 'c'=='configurable' and 'w'=='writable':

```
prf(obj, 'foo', 'bar', 'ew') //
enumerable and writable but not
configurable
// multiple properties:
prf(obj, { one: 'one', two: 'two' }, 'ewc') // configurable too
```

## Where can I use it?

Anywhere! For pre-ES5 environments *prf* will simply fall-back to an `object[property] = value` so you can get close to what you want.

*prf* is Ender-compatible so you can include it in your Ender build and `$.prf(...)` or `var prf = require('prf'); prf(...).`

## Licence

*prf* is Copyright (c) 2013 Rod Vagg [@rvagg](#) and licensed under the MIT licence. All rights not explicitly granted in the MIT license are reserved. See the included LICENCE.md file for more details.