

Interface: Options

Properties

circles?

```
optional circles: boolean
```

Keeping track of circular references will slow down performance with an additional 25% overhead. Even if an object doesn't have any circular references, the tracking overhead is the cost. By default if an object with a circular reference is passed to `rfdc`, it will throw (similar to how `JSON.stringify` would throw).

Use the `circles` option to detect and preserve circular references in the object. If performance is important, try removing the circular reference from the object (set to `undefined`) and then add it back manually after cloning instead of using this option.

proto?

```
optional proto: boolean
```

Copy prototype properties as well as own properties into the new object.

It's marginally faster to allow enumerable properties on the prototype to be copied into the cloned object (not onto it's prototype, directly onto the object).

To explain by way of code:

```
require('rfdc')({ proto: false })(Object.create({a: 1})) // => {}  
require('rfdc')({ proto: true })(Object.create({a: 1})) // => {a: 1}
```

Setting `proto` to `true` will provide an additional 2% performance boost.