

Deferreds

Unit 8

Warning!

Callbacks

Callbacks

- Generic way of saying “This is what I want you to do when you are done with your asynch operations”


```
$.get("something.html", function() {  
    console.log("I run in the future!");  
});
```


Lets kick it up a notch!


```
$.get("something.html", function() {  
    $.get("somethingelse.html", function() {  
        console.log("ok now I'm really done");  
    })  
});
```



```
var r1=false;
var r2=false;
$.get("something.html", function() {
    r1=true;
    if(r2) complete();
});
```

```
$.get("somethingelse.html", function() {
    r2=true;
    if(r1) complete();
});
```


Enter the Deferred

Deferreds represent
Async values


```
content = somethingAsync();  
console.log(content); //Error!
```



```
contentDeferred = somethingBetterAsync();  
contentDeferred.done(function(stuff) {  
    //I'm done!  
});
```



```
contentDeferred = somethingAsync();  
somethingDeferred = somethingElseAsync();  
  
$.when(contentDeferred,somethingDeferred).done(  
function(res1, res2) {  
  
});
```


Deferreds

Deferreds

- An *asynch* operation

Deferreds

- An asynch operation
- Your code can resolve or reject them

Deferreds

- An asynch operation
- Your code can resolve or reject them
- Have 3 events: Done, Fail, and Always

Promises

- Read Only Deferred
- Basically, you can't resolve or reject, just listen for done, fail, and always
- jQuery Ajax functions return promises

BTW

- This is all rolling into JavaScript natively
- ES6 (ECMAScript 6)
- They are a bit different

Demos

Making your own Deferreds

- For other async operations not covered by jQuery (IndexedDB, WebSQL)
- For things that **may** be async

Demos