|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Async foundations | callbacks  Higher order functions are functions that accept a callback as a parameter | Callback is basically a function itself | Callback example:  Function callback(){  Console.log(“coming from callback”);  }  Function higherOrder(fn){  Console.log(“about to call callback”);  fn();  console.log(“callback has been invoked”);  }  higherOrder(callback); | Uses of callbacks:   1. Advanced array methods 2. Browser events 3. Ajax requests 4. React development |
| Use an anonymous function as callback instead of function declaration,  Function greet(  “john”,function(name){  return name.toUpperCase();  }  ); | Callback with function decalation:  Function greet(name,formatter){  Return “hello, “+formatter(name);  }  Function upperCaseName(name){  Return name.toUpperCase();  }  Greet(“john”,upperCaseName); | foreach  var arr=[1,2,3,4,5]  for(var i;i<arr.length;i++){  console.log(arr[i]\*2);  }  Double(arr)  Define foreach function  Function foreach(arr,callback){  for(var i=0;i<arr.length;i++){  callback(arr[i],I,arr);  }  } | Var arr=[“my”,”name”,”is”,”john”]  Var Result=””;  Foreach(arr,function(str,index,arr){  If(arr.length-1!==index){  Result+= str+” ”;  Else(  Result+=str+”!!!”;  )  }  });  Console.log(Result) |
| Findindex  Findindex returns the index of the first element in the array for which the callback returns a truthy value.-1 is returned when the callback never ereturns a truthy value.  Stack is an ordered set of stackframes. The bottom of the stack is where the first ink=voked function lives. | Stack and heap  Stack is an ordered data structure.  It keeps track of function invocations.  When a function is invoked, the stack gets modified to keep track of that function you invoke and when the invocation is done the functione gets poped off the stack and the javascript runtime will take itoff for you. | When a function is invoked, it gets pushed to the top of the stack and when the function has returned, it gets popped off the top. | Stackframe contains   1. Invked function 2. Params passed to the function 3. Current line number |
| Heap is an area in memory where data is stored. | Function uppercasefirst(word){  Return word[0].toUpperCase()+word.slice(1);  }  Function titlecase(sentence){  Words=sentence.slice(“ ”);  For (var i=0;i<words.length;i++){  Return words[0].toUpperCase()+words.slice(1);  }  Return words.join(“ “);  }  Titlecase(sentence); | **How the stack works for that code**   |  | | --- | |  | | join | | slice | | uppercase | | uppercasefirst | | slice | | titlecase | | main |   [] indicates functions that have come up and gone off while program is still running  [] indicates functions that have come up and gone off while program is done running | SetTimeout and setInterval |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |