

Context Quiz Recall

```
let cat = {  
  name: "Jet",  
  sayName: function() {  
    return this.name;  
  }  
};  
  
console.log(cat.sayName()); // ???
```

What is the returned value from the above invocation of `cat.sayName`?

- ☐ undefined
- ☒ Jet
- ☐ the global object
- ☐ object
- ☐ cat

EXPLANATION

The `cat` object is the context of the `cat.sayName` method-style invocation so the returned value will be `cat.name`- which is "Jet".

```
let panther = {  
  pounce: function() {  
    console.log("woosh");  
  },  
  hunt: function() {  
    this.pounce();  
  }  
}
```

```
};
```

```
let goHunt = panther.hunt;  
goHunt(); // TypeError: this.pounce is not a function
```

What is the context of the above `goHunt` function?

☒ the global object

☐ `panther`

☐ `hunt`

☐ `object`

EXPLANATION

When we extract the `goHunt` method to a separate variable and then try to invoke it - the `goHunt` will have lost the context of the `panther` object. So the `goHunt` will instead be called upon the global object.

```
function sayThis() {  
  if (true) {  
    console.log(this); // => ???  
  }  
}  
  
sayThis();
```

What is the value printed when we invoke the `sayThis` function above in Node?

☒ the global object

☐ `sayThis`

- ☐ block
- ☐ undefined
- ☐ None of the above

EXPLANATION

The global object is the context for every function call that does not have another defined context.

A ____ is a function that is a value within an object and belongs to that object.

- ☐ this
- ☒ method
- ☐ object
- ☐ None of the above
- ☐ context

EXPLANATION

A *method* is a function that is a value within an object and belongs to an object.

```
let panther = {  
  pounce: function() {  
    console.log("woosh");  
  },  
  hunt: function() {  
    this.pounce();  
  }  
};
```

```
let boundHunt = panther.hunt.bind(panther);
```

What is the context of invoking the `boundHunt` function above?

- ☐ `object`
- ☒ `panther`
- ☐ `hunt`
- ☐ the global object

EXPLANATION

When we extract the `goHunt` method and *bind* it to the `panther` object then no matter where `boundHunt` is called it will have the bound context of the `panther` object.

```
let cat = {  
  whoIsThis: function() {  
    return this;  
  }  
};  
  
console.log(cat.whoIsThis()); // ???
```

What is the returned context from the above invocation of `cat.whoIsThis`?

- ☐ `undefined`
- ☒ `cat`
- ☐ the global object
- ☐ `object`

☐ Jet

EXPLANATION

The `cat` object is the context of the `cat.whoIsThis` method-style invocation.

The value of `this` in a function is the function's ____.

☐ scope

☐ method

☐ None of the above

☒ context

☐ `this`

EXPLANATION

The value of `this` in a function is that function's context.