

## 浙江大学城市学院实验报告

课程名称 跨平台脚本开发技术

实验项目名称 实验十一 HTTP 协议

学生姓名 吴成洋 专业班级 软件工程 1404 学号 31401417

实验成绩            指导老师（签名）                      日期                     

注意：

- 务请保存好各自的源代码，以备后用。
- 请把作业保存为 pdf 上传到 BB 平台，请务必在截止日期前提交。

### 实验目的：

- 1、掌握 JS 中的模块定义与使用。
- 2、了解编程语言及其解释器的原理与实现

### 实验内容：

1. 阅读课件讲义并运行讲义示例程序
2. 教材 P248 习题 17.13.1，17.13.2

### 实验步骤：

1、

已阅读相关代码。

2、

2.1、

```

text/plain:
My name is Marijn Haverbeke. You can email me at marijn@haverbeke.nl, or visit my website, http://marijnhaverbeke.nl .

text/html:
<!doctype html>

<div vocab="http://schema.org/" typeof="Person">
  <p><span property="name">Marijn Haverbeke</span>, <span
    property="jobTitle">Programmer</span></p>

  <p>You can reach me at <a href="mailto:marijn@haverbeke.nl"
    property="email">marijn@haverbeke.nl</a>, or visit my web page, <a
      href="http://marijnhaverbeke.nl" property="url">marijnhaverbeke.nl</a>.</p>
</div>

application/json:
{
  "name": "Marijn Haverbeke",
  "email": "marijn@haverbeke.nl",
  "website": "http://marijnhaverbeke.nl"
}

application/rainbows+unicorns:

```

The image shows a large, stylized ASCII art representation of a unicorn's head. The unicorn is facing right. Its mane is composed of various rainbow-colored characters like 'r', 'o', 'n', 'i', 's', 'k', 'y'. Its body and tail are also decorated with similar colorful characters. The entire artwork is created using a combination of different colored fonts.

代码如下

```
function getContent(content){  
    var req = new XMLHttpRequest();  
    req.open("GET" , "http://eloquentjavascript.net/author" , false);  
    req.setRequestHeader("accept" , content);  
    req.send(null);  
    return req.responseText;  
}  
  
var contents = ["text/plain" ,  
                "text/html" ,
```

```
        "application/json" ,  
        "application/rainbows+unicorns"];  
  
contents.forEach(function(content){  
    try {  
        console.log( content + ":\n" + getcontent(content) , "\n");  
    } catch (error) {  
        console.log("Error:" + error);  
    }  
});
```

## 2.2、

```
This should be []: ▶ []  
◀ ▶ Promise {[[PromiseStatus]]: "resolved", [[PromiseValue]]: undefined}  
This should be [1, 2, 3]: ▶ (3) [1, 2, 3]  
>
```

代码如下：

```
function all(promises){  
    return new Promise(function(succeed,fail){  
        var results = [];  
        var pending = promises.length;  
        promises.forEach(function(promise , i){  
            promise.then(function(result){  
                results[i] =result;  
                pending -= 1;  
                if(pending == 0 )  
                    succeed(results);  
            },function(error){
```

```
        fail(error);
    });
});
if(promises.length == 0){
    succeed(results);
}
});
}

// Test code.
all([]).then(function(array) {
    console.log("This should be []:", array);
});
function soon(val) {
    return new Promise(function(success) {
        setTimeout(function() { success(val); },
            Math.random() * 500);
    });
}
all([soon(1), soon(2), soon(3)]).then(function(array) {
    console.log("This should be [1, 2, 3]:", array);
});
function fail() {
    return new Promise(function(success, fail) {
        fail(new Error("boom"));
    });
}
all([soon(1), fail(), soon(3)]).then(function(array) {
    console.log("We should not get here");
}, function(error) {
    if (error.message != "boom")
        console.log("Unexpected failure:", error);
});
```