

# 轮播组件开发

## 极客大学前端训练营

### 线上 Week 01

程劭非 (winter)

前手机淘宝前端负责人

# Extra：编程练习

# 组件化

# 组件化的起源

`<div> => <my-component>`

# 组件化的核心目标

- 复用：组件将会作为一种复用单元，被用在多处。
- 解耦：组件本身隔离了变化，组件开发者和业务开发者可以根据组件的约定各自独立开发和测试。
- 封装：组件屏蔽了内部的细节，组件的使用者可以只关心组件的属性和事件和方法。
- 抽象：组件通过属性和事件、方法等基础设施提供了一种描述UI的统一模式，降低了使用者学习的心智成本。

# 互动：如何定义一个组件

# 组件化的基本思路

```
var element = document.createElement("div");  
document.getElementById("container").appendChild(element);
```

# 组件化的基本思路

```
function MyComponent(){  
  this.prop1;  
  this.method1;  
  .....  
}
```



# 组件化的基本思路

```
function MyComponent(){  
  this.root = document.createElement("div");  
  this.appendTo = function(node){  
    node.appendChild(this._root)  
  }  
}
```

# 组件化的基本思路

```
function MyComponent(){
  var _root = document.createElement("div");
  root.prop1 // = ...
  root.method1 = function(){
    /*.....*/

  }
  return root;
}
document.getElementById("container").appendChild(new MyComponent());
```

# 组件化的发展



# 三大框架组件方案的演示

# 实际操作

设计一个轮播组件

- 排版布局
- 动画
- 拖拽



# 轮播组件的需求分析

- 拖拽回弹的行为研究
- 动画的曲线
- 点击与拖拽
- 父元素的行为

THANKS! |  极客大学