

闭包是什么，闭包会用在哪里？

- 函数作为返回值
- 函数作为参数

```
// 函数作为返回值
function create() {
  let a = 100;
  return function() {
    console.log(a);
  }
}
let fn = create();
let a = 200;
fn(); // 100
```

```
// 函数作为参数
function print(fn) {
  let a = 200;
  fn();
}
let a = 100;
function fn() {
  console.log(a)
}
print(fn); // 100
```

所有的自由变量的查找，是在函数定义的地方，向上级作用域查找，不是在执行的地方。

连环问：闭包的使用场景？

隐藏数据，只提供 API 访问，比如做一个简单的 cache 工具。

```
// 做一个简单的 cache 工具
function createCache() {
  let data = {};
  return {
    get: function(key) {
      return data[key];
    },
    set: function(key, value) {
      data[key] = value;
    }
  }
}
const c = createCache();
```

```
c.set("a", 100);
c.get("a");
// 在这一层访问不到 data, 只能访问到 data 返回的数据
```

连环问：创建 10 个 a 标签，点击弹出结果

```
// 创建 10 个 a 标签，点击弹出结果
let i, a;
for (i = 0; i < 10; i++) {
  const a = document.createElement("a");
  a.innerHTML = i + "<br />";
  a.addEventListener("click", function (e) {
    e.preventDefault();
    alert(i);
  });
  document.body.appendChild(a);
}

// 这种情况不论点击哪一个数，都是弹出 10。
```

上面代码，不论点哪一个数都是弹出 10，是因为 `i` 是全局作用域的变量，在执行点击事件的时候才会弹出 `i` 的值，而这个时候 `i` 已经经过一遍循环了，`i` 已经变成了 10。所以弹出 10。

```
let a;
for (let i = 0; i < 10; i++) {
  const a = document.createElement("a");
  a.innerHTML = i + "<br />";
  a.addEventListener("click", function (e) {
    e.preventDefault();
    alert(i);
  });
  document.body.appendChild(a);
}
```

上面代码，点击 1 就弹出 1，点击 2 就弹出 2。

是因为 `i` 是块级级作用域，只在 `for` 循环体内有效，在 `for` 循环块儿内，每循环一次，`i` 的值就变一次。