

```
Array.prototype.unshiftCustom = function () {
  let unshiftLength = arguments.length;

  for (let i = this.length + unshiftLength - 1; i > unshiftLength - 1; i--) {
    this[i] = this[i - unshiftLength];
  }

  for (let i = 0; i < unshiftLength; i++) {
    this[i] = arguments[i];
  }

  return this.length;
}
```

```
Array.prototype.unshiftCustom = function () {
  // 注意下方的 concat, 要规避 this 是非标准数组的情况
  // 以后凡是遇到 concat 都需要注意该问题
  // let temp = [].concat(this);
  let temp = this.map(function(v) {
    return v;
  });

  this.length = 0;

  for (let i = 0; i < arguments.length; i++) {
    this[this.length] = arguments[i];
  }

  for (let i = 0; i < temp.length; i++) {
    this[this.length] = temp[i];
  }

  return this.length;
}
```

```
Array.prototype.unshiftCustom = function () {
  // 如果你实现了 splice, 也可以直接使用下面的方式
  // 但是, 实际上 unshift 方法的实现方式要比 splice 简单很多, 使用 splice 并不是一个好方式
  // 而且该方法使用了 es6 的扩展运算符, 兼容性很差
  Array.prototype.splice.apply(this, [0, 0, ...arguments]);

  return this.length;
}
```