只放页码组件

```
<template>
  <div class="pagination">
    <ul>
      <li class="page-prev" :class="{'disabled': current == 1}"><a href="javascript:;"
@click="setCurrent(current - 1)"> <img src="../assets/images/syy.png" alt=""/></a></li>

      <li v-for="p in grouplist" :class="{'curr': current == p.val}"><a href="javascript:;"

@click="setCurrent(p.val)"> {{ p.text }} </a>
      </li>
      <li class="page-next" :class="{'disabled': current == page}"><a href="javascript:;"
@click="setCurrent(current + 1)"><img src="../assets/images/xyy.png" alt=""/></a></li>
    </ul>
  </div>
</template>

<script>
  export default {
    name: '',
    data () {
      return {
        current:this.currentPage
      }
    },
    props: {
      page: {// 总页数
        type: Number,
        default: 0
      },
      currentPage: {// 当前页码
        type: Number,
        default: 1
      },
      pagegroup: {// 分页条数
        type: Number,
        default: 5,
        coerce: function (v) {
          v = v > 0 ? v : 5;
          return v % 2 === 1 ? v : v + 1;
        }
      }
    },
    computed: {
    /*  page: function () { // 总页数
        return ;
      },*/
      grouplist: function () { // 获取分页页码
        var len = this.page, temp = [], list = [], count = Math.floor(this.pagegroup / 2),
center = this.current;
        if (len <= this.pagegroup) {
          while (len--) {
            temp.push({text: this.page - len, val: this.page - len});
          }
          ;
          return temp;
        }
        while (len--) {
          temp.push(this.page - len);
        }
        ;
        var idx = temp.indexOf(center);
        (idx < count) && ( center = center + count - idx);
        (this.current > this.page - count) && ( center = this.page - count);
        temp = temp.splice(center - count - 1, this.pagegroup);
        do {
          var t = temp.shift();
          list.push({
            text: t,
            val: t
          });
        } while (temp.length);
        if (this.page > this.pagegroup) {
          (this.current > count + 1) && list.unshift({text: ’...’, val: list[0].val - 1});
          (this.current < this.page - count) && list.push({text: ’...’, val:
list[list.length - 1].val + 1});
        }
        return list;
      }
    },
    methods: {
      setCurrent: function (idx) {
        if (this.current != idx && idx > 0 && idx < this.page + 1) {
          this.current = idx;
          this.$emit('pagechange', this.current);
        }
      }
    }
  }
</script>

<!-- Add "scoped" attribute to limit CSS to this component only -->
```

调用页码的组件

页码位置

```
<pagination :page="page" :current-page="current" :pagegroup="pagegroup" @pagechange="pagechange"></pagination>
```

```
methods:{
  pagechange: function (currentPage) {
    console.log(currentPage);
    this.$http.get('http://yuntai.yn4.chuanhai.net/api/web/v1/counselor/counselor', {
      headers: {
        JOKE: 'chuanhai!',
        Device: 'deviceunique'
      },
      params:{
        page:currentPage
      }
    })
      .then((response)=>{
        console.log(response);
        this.newsLists = response.body.res;
        this.page=response.totalPage;
        console.log(this.page)
      })
      .catch((error)=>{
        console.log(error)
      })
  }
},
created(){
  this.$http.get(’http://yuntai.yn4.chuanhai.net/api/web/v1/counselor/counselor’, {
    headers: {
      JOKE: 'chuanhai!’,
      Device: 'deviceunique’
    },
    params:{
      page:1
    }
  })
    .then((response)=>{
      console.log(response);
      this.newsLists = response.body.res;
      this.page=response.body.totalPage;
      console.log(this.page)
    })
    .catch((error)=>{
      console.log(error)
    })
}
```