

Gémissement

Ceci est le fichier README.md du projet github <https://github.com/lzwjava/Creak>.

Creak

Analyseur HTML élégant en Swift.

Exemple

```
import Foundation
import XCTest

@testable import Creak

class HtmlNodeTest: BaseTest {

    func testInnerHtml() {
        let div = Tag(name: "div")
        div.setAttributes(["class": AttrValue("all")])
        let a = Tag(name: "a")
        a.setAttributes(["href": AttrValue("http://google.com", doubleQuote: false)])
        let br = Tag(name: "br")
        br.selfClosing = true

        let parent = HtmlNode(tag: div)
        let childa = HtmlNode(tag: a)
        let childbr = HtmlNode(tag: br)
        parent.addChild(childa)
        parent.addChild(childbr)
        childa.addChild(TextNode(text: "link"))

        XCTAssertEqual("<a href='http://google.com'>link</a><br />", parent.innerHTML())

        // test cache
        XCTAssertEqual("<a href='http://google.com'>link</a><br />", parent.innerHTML())
    }
}
```

```
}
```

```
func testOuterHtml() {
    let div = Tag(name: "div")
    div.setAttributes(["class": AttrValue("all")])
    let a = Tag(name: "a")
    a.setAttributes(["href": AttrValue("http://google.com", doubleQuote: false)])
    let br = Tag(name: "br")
    br.selfClosing = true

    let parent = HtmlNode(tag: div)
    let childa = HtmlNode(tag: a)
    let childbr = HtmlNode(tag: br)
    parent.addChild(childa)
    parent.addChild(childbr)
    childa.addChild(TextNode(text: "link"))

    XCTAssertEqual("<div class=\"all\"><a href='http://google.com'>link</a><br /></div>", parent.outerHtml())

    // test cache
    XCTAssertEqual("<div class=\"all\"><a href='http://google.com'>link</a><br /></div>", parent.outerHtml())
}
```

```
func testOuterHtmlEmpty() {
    let a = Tag(name: "a")
    a.setAttributes(["href": AttrValue("http://google.com", doubleQuote: false)])
    let node = HtmlNode(tag: a)
    XCTAssertEqual("<a href='http://google.com'></a>", node.outerHtml())
}
```

```
func testOuterHtmlNoValueAttribute() {
    let div = Tag(name: "div")
    div.setAttributes(["class": AttrValue("all")])
    let a = Tag(name: "a")
    a.setAttributes(["href": AttrValue("http://google.com", doubleQuote: false)])
    let br = Tag(name: "br")
    br.selfClosing = true

    let parent = HtmlNode(tag: div)
    let childa = HtmlNode(tag: a)
```

```

        childa.setAttribute("ui-view", attrValue: AttrValue(nil))
        let childbr = HtmlNode(tag: br)
        parent.addChild(childa)
        parent.addChild(childbr)
        childa.addChild(TextNode(text: "link"))

        XCTAssertEqual("<div class=\"all\"><a href='http://google.com' ui-view>link</a><br /></div>", parent.text())
    }

    func testText() {
        let a = Tag(name: "a")
        let node = HtmlNode(tag: a)
        node.addChild(TextNode(text: "link"))
        XCTAssertEqual("link", node.text())

        // test cache
        XCTAssertEqual("link", node.text())
    }

    func testTextLookInChildren() {
        let p = HtmlNode(tag: "p")
        let a = HtmlNode(tag: "a")
        p.addChild(TextNode(text: "Please "))
        a.addChild(TextNode(text: "click me"))
        p.addChild(a)
        p.addChild(TextNode(text: "!"))
        XCTAssertEqual("Please click me!", p.textWithChildren(true))
    }

    func testGetAttribute() {
        let node = HtmlNode(tag: "a")
        node.tag.setAttributes([
            "href": AttrValue("http://google.com", doubleQuote: false),
            "class": AttrValue("outerlink rounded", doubleQuote: true)])
        XCTAssertEqual("outerlink rounded", node.attribute("class"))
    }
}

```