

Joke Test Chuck Norris

Ciao Sandro,

Per provare il Test del Joke basta fare un doppio Click al file index.html che si trova nella cartella dist del progetto :-)

Questa è la url per il download del progetto da Github:

Https: <https://github.com/giooss13/giooss13.git>

Ssh: [git@github.com:giooss13/giooss13.git](ssh://git@github.com:giooss13/giooss13.git)

Github Cli: `gh repo clone giooss13/giooss13`

Init del progetto:

Nel progetto ho scelto come prima configurazione quella di Gulp e non Webpack sostanzialmente perché un progetto piccolino, ed un task runner come Gulp era più adatto appunto.

Workflow:

Per quanto riguarda il sistema di versionamento ho usato Git ed essendo l'unico dev ho lavorato su ramo secondario ed una volta finito ho portato il tutto sul branch Master.

Scelta style, script :

Ho deciso di usare per lo style la sintassi scss e una volta finito compilarlo nel classico style.css, mi sono permesso di aggiungere delle piccole animazioni anche se non richiesto il tutto puro e senza plugin, il tutto responsive tramite flex box grid e con unita di misura px e vw

Ho deciso per il js di usare vanilla puro in quanto mi veniva più facile scrivere rispetto ad un jQuery, secondo me perché più leggibile sintatticamente:

Ho variabilizzato gli elementi di Markup per una scrittura più veloce e ordinata

- Ho pensato di usare la chiamata asincrona fetch sia per l'endpoint randomico che quello di ricerca, perché più moderna e versatile rispetto ad una Ajax (in jquery): la promise non andrà in rigetto sullo stato di errore HTTP anche se la risposta è un HTTP 404 o 500. E non appena il server risponde con le intestazioni richieste, la Promise si risolverà normalmente.
- Fetch di ricerca - searchFunc(): questa chiamata è implementato con Metodo Get accettando come parametri gli HEADERS della chiamata, Json come in questo in questo caso
- La risposta (res) accetta come parametri un json che viene poi a sua volta istanziato come oggetto e poi stampato all'occorrenza con l'opportuno target.
- funzione e fetch annessa, ckLoop () Nel caso della randomica, simile a quella di ricerca (ovviamente con endPoint diverso) la Res viene stampata ogni 3 secondi perché racchiusa in una funzione Loop chiamata da me CK

spero di essere stato chiaro e di aver soddisfatto in parte quel che da me ti aspettavi,

Grazie mille

Giovanni