

## CODE REVIEW

### Type Script

TypeScript és un superset de JavaScript desenvolupat per Microsoft que afegeix tipatge estàtic opcional. Això permet detectar errors durant la fase de desenvolupament, abans d'executar el codi. Compila a JavaScript estàndard, assegurant compatibilitat amb tots els navegadors i entorns JavaScript.

### Baixar codi

1. Instal·la Node utilitzant la versió que vas descarregar de nodejs.org.
2. Obre un terminal.
3. Fes un fork (la teva pròpia còpia) de TypeScript al teu compte de GitHub, després fes un clon (una còpia local) al teu ordinador.  
**git clone https://github.com/microsoft/TypeScript.git**  
Afegeix **--depth=1** al final de la comanda git clone per estalviar temps.
4. Instal·la l'eina de línia de comandes hereby: **npm install -g hereby**
5. Canvia al directori de TypeScript que has creat: **cd TypeScript**
6. Instal·la les dependències: **npm ci**
7. Assegura't que tot es construeix i que passen els tests: **hereby runtests-parallel**
8. Obre el directori de TypeScript en el teu editor.
9. Segueix les instruccions a continuació per afegir i depurar un test.

Per baixar el codi, Github permet descarregar-lo i proporciona instruccions sobre la comanda git clone, entre altres.

### Escriure/millorar/resoldre bugs del codi

1. Clonar el codi del repositori de GitHub al teu ordinador.
2. Analitzar el codi i utilitzar eines com la consola de depuració o un depurador per identificar la causa de l'error.
3. Desenvolupar una solució per l'error, corregint errors de sintaxi o lògica, o modificant l'estructura del codi si cal.
4. Comprovar que la solució proposta resol l'error executant el codi i validant que el comportament sigui l'esperat.
5. Si treballes en un projecte col·laboratiu, crear una nova branca al teu repositori local per implementar els canvis necessaris.
6. Després de provar i verificar que la solució funciona, iniciar una sol·licitud d'extracció per fusionar els canvis al repositori principal.
7. Un altre membre de l'equip haurà de revisar i aprovar la sol·licitud d'extracció abans que els canvis es fusionin amb el repositori principal.

1. **Crea una nova branca** per treballar en els teus canvis:

```
git checkout -b feature/nova-funcionalitat
```

2. **Realitza els canvis** en el codi font. Els arxius principals es troben a la carpeta ``src``.
3. **Executa els tests** per assegurar-te que els canvis no introdueixen errors:  
`npm test`

Pots afegir o modificar tests per cobrir els teus canvis a la carpeta ``tests``.

Per resoldre bugs i millorar el codi, GitHub disposa d'una interfície on es poden identificar els bugs que cal solucionar i proposar millores. També ofereix explicacions sobre com fer-ho.

## PUJAR CODI

Per pujar el codi, has de seguir aquests passos:

1. Enllaç al problema: Quan enviïs una pull request (PR), assegura't d'incloure un enllaç a l'issue o problema que estàs intentant solucionar. Si ja has enviat una PR per a aquest problema, deixa un comentari a l'issue per evitar que altres persones intentin solucionar-lo de nou.
2. Descripció del canvi: Proporciona una descripció clara i detallada del que fa el teu canvi. Això ajudarà els revisors a entendre l'objectiu i l'abast de la teva PR.
3. Basar-se en el codi recent: Abans de fer el commit dels teus canvis, assegura't d'actualitzar la teva branca amb la versió més recent del codi principal (normalment la branca `main` o `master`).
4. Incloure tests: Afegeix tests adequats per assegurar-te que el teu canvi funcioni correctament.
5. Verificació de tests: Assegura't que almenys un dels tests falli quan el teu canvi no s'aplica. Si la teva PR no compleix aquest criteri, explica les raons en la descripció de la PR.
6. Cobertura de tests: Els tests haurien de comprovar diferents combinacions i casos límit del teu canvi per assegurar que funcioni en diverses situacions.
7. Incorporar canvis de base: Inclou els canvis de base (rebase) amb els teus canvis abans de fer el push.
8. Normes de codificació: Segueix les normes de codificació establertes en les Coding Guidelines del repositori de TypeScript.
9. Configuració de git: Configura les opcions `autocrlf` i `whitespace` de git per evitar problemes amb els caràcters de final de línia. Pots fer-ho amb les següents comandes:

```
git config --global core.autocrlf input
```

```
git config --global core.whitespace cr-at-eol
```

Per pujar el codi, GitHub disposa dels pull requests, on pots especificar el teu canvi i els tests que demostren que funciona, entre altres coses.