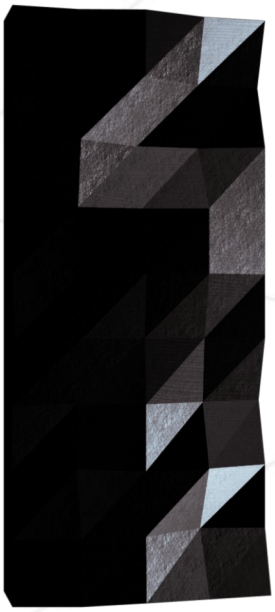

42 standardı

Version 2.0.0

Mathieu mathieu@staff.42.fr
Gaëtan gaetan@staff.42.fr

Özet: bu belge 42'de yürürlükte olan C standardını açıklamaktadır. Bir standart programlama, kod yazmayı düzenleyen bir dizi kural tanımlar. O C'den 42'ye yazarken standarda uymak zorunludur

İçindekiler tablosu

Bölüm I

Önsöz

Bu belge, 42'de yürürlükte olan standart C'yi açıklamaktadır. Bir programlama standardı, kod yazmayı düzenleyen bir dizi kural tanımlar. C'den 42'ye yazarken standarda uymak zorunludur.

I.1 Neden bir standart uyguluyorsunuz ?

Standardın iki temel amacı vardır: kodlarınızı herkesin kolayca okuyabilmesi için standartlaştırmak, öğrenciler ve denetçiler. Basit ve net kodlar yazın.

I.2 Renderlerinizde standart

Tüm C kod dosyalarınız 42 standardına uygun olmalıdır. Standart, düzelticileriniz tarafından doğrulanacak ve en ufak bir standart eksikliği, projenize veya alıştırmanıza 0 Puan verecektir. Eş düzeltmeler sırasında düzelticiniz, işleminizde "Norminette" yi başlatmak zorunda kalacak, yalnızca" Norminette " nin sonucu dikkate alınmalıdır.

I.3 Tavsiye

Çabucak anlayacağınız gibi, norm bir kısıtlama değildir. Bu aksine, standart, basit bir C yazarken size rehberlik edecek bir korkuluktur ve temel. Doğrudan standart kodu için çok önemli olmasının nedeni budur , q ve ardından ilk birkaç saat içinde daha yavaş kodlama. İçeren bir kaynak dosya standart bir hata, on olan bir dosya kadar kötüdür. Çalışkan ve uygulandı ve standart kısa sürede bir otomatizm haline gelecektir.

I.4 Sorumluluk Reddi

Hatalar mutlaka "Norminette" de var, lütfen bunları bölümde bildirin forum içi. Yine de,"Norminette" orijinaldir ve renderleriniz uyum sağlamalıdır böcekler onun için:).

Bölüm II

42 standardı

II.1 Adlandırma kuralı

Zorunlu kısım

- Bir struct adı s_ ile başlamalıdır.
- Bir typedef adı t_ ile başlamalıdır.
- Bir union adı u_ ile başlamalıdır.
- Enum adı e_ ile başlamalıdır.
- Global bir ad g_ ile başlamalıdır.
- değişkenlerin ve işlevlerin adları yalnızca küçük harf, sayı ve ' _ ' alt tire den oluşmalıdır.
- Dosya adları ve dizinler yalnızca aşağıdakilerden oluşmalıdır
  küçük harf, sayılar ve ' _ ' alt tire.
- Dosya derlenebilir olmalıdır.
- Ascıı standart tablosunun bir parçası olmayan karakterlere izin verilmez.

Önerilen bölüm

- (Değişkenler, fonksiyonlar, makrolar, türleri, dosyaları veya dizinleri) nesneler olmalı
  en açık veya anımsatıcı isimler. Sadece 'bilgisayarlar' olabilir
  istediğiniz gibi atandı
- Kısaltmalar, ismin boyutunun anlamını kaybetmeden anlamlı bir şekilde azaltılmasına izin verdiği ölçüde tolere edilir. Bileşik isimlerin bölümleri '_'ile ayrılacaktır.
- Tüm tanımlayıcılar (işlevler, makrolar, türler, değişkenler vb.) İngilizce olmalıdır.
- Genel değişkenlerin herhangi bir kullanımı gerekçelendirilmelidir.

II.2 Biçimlendirir

Zorunlu kısım

- Tüm dosyalarınızın ilkinden itibaren standart 42 başlığıyla başlaması gerekir sıra.
- Her işlev, parantezleri saymadan en fazla 25 satır olmalıdır. işlev bloğu.
- Her satır, yorum da dahil olmak üzere en fazla 80 sütun olabilir. Dikkat : sekme bir sütun için değil, içerdiği n boşluk için sayılır gösteriyordu.
- Satır başına yalnızca bir yönerge.
- Boş bir satır boşluk veya sekme içermemelidir.
- Bir satır hiçbir zaman boşluk veya sekme ile bitmemelidir.
- Bir açılış veya kapanış ayracı veya yapının sonu ile karşılaştığınızda kontrol sizde, çizgiye geri dönmelisiniz.
- Ayrıca kodunuzu 4 boşluklu sekmelerle girintilemeniz gerekir. (Öyle değil 4 boşluğa eşdeğer değildir, bunlar gerçekten sekmelerdir.)
- Değilsek her virgül veya noktalı virgülden sonra bir boşluk gelmelidir. satırın sonunda.
- Her operatör (ikili veya üçlü) ve işlenen bir uzay ve tek.
- Her C anahtar sözcüğünden sonra, tür anahtar sözcükleri dışında bir boşluk gelmelidir.değişken (int, char, float, vb. gibi) ve sizeof. (Tuttuk çünkü KRP, tesadüfen çift olan 82 dedi)
- Her değişken bildirimi aynı sütunda girintili olmalıdır.
- İşaretçilerin yıldızları değişkenin adına yapıştırılmalıdır.
- Satır başına yalnızca bir değişken bildirimi.
- Global değişkenler ve statik değişkenler dışında aynı satırda bir bildirim ve bir örnekleme yapamazsınız.
- Bildirimler, işlevlerin başında olmalı ve işlevlerden ayrılmalıdır. boş bir satır ile uygulama.
- Bildirimlerin veya uygulamanın ortasında boş satırlar bulunmamalıdır.
- Aynı talimat veya yapı sırasında yeni bir satıra dönebilirsiniz. kontrol, ancak parantez veya operatör ile bir girinti eklemelisiniz atama. Operatörler hattın başında olmalıdır.

```
toto = 42 + 5
   - 28;             //DOĞRU (Ama okuması zor :P )

if (toto == 0
   && titi == 2
   && (tutu == 3                //DOĞRU (Ama okuması zor :P )
       || tata == 4)
   || rara == 3)
```

- Çoklu atama yasaktır.

II.3 Fonksiyon Parametreleri

Zorunlu kısım

- Bir Fonksiyon en fazla 4 adlandırılmış parametre alır.
- Argüman almayan bir fonksiyon ile açıkça prototip oluşturulmalıdır. bir argüman olarak geçersiz kelime.

II.4 Fonksiynlar

Zorunlu kısım

- Fonksiyon prototiplerinin değişken adları olmalıdır.
- Her fonksiyon tanımı boş bir satırla ayrılmalıdır.

Önerilen parça

- Fonksiyon isimleri aynı dosya içinde aynı hizada olmalıdır. Bu geçerlidir headers.

II.5 Typedef, struct, enum ve union

Partie obligatoire

- Struct, enum ve union bildirirken bir sekme ayarlamanız gerekir.
- Struct, enum veya union türünde bir değişken bildirirken koymayacaksınız tipteki bir boşluktan daha fazla.
- Bir typedef iki parametre arasında bir sekme kullanmanız gerekir.
- Bir typedef ile struct, union veya enum bildirdiğinizde, tüm kurallar Uygula ve typedef adını struct adıyla hizalamanız gerekir, union veya enum.
- Dosyadaki bir yapıyı bildiremezsiniz .c.

II.6 Header lar

Zorunlu kısım

- Yalnızca headers (sistem veya değil), tanımlar, tanımlar, başlık dosyalarında prototiplere ve makrolara izin verilir.
- h'nin tüm içeriği dosyanın (.c veya .h) başında yapılmalıdır.
- Header ları çift dahil edilmeye karşı koruyacağız. Dosya ft_foo.h ise,la makro tanık "FT_FOO_H".
- fonksiyonların prototipleri yalnızca .h dosyalarında bulunmalıdır.
- Kullanılmayan header (.h) eklenmesi yasaktır.

Önerilen parça

- Herhangi bir header ın dahil edilmesi, .c'de olduğu kadar .h'de de gerekçelendirilmelidir.

II.7 Makrolar ve preprocessor

Zorunlu kısım

- Tanımlayıcı kod tanımlamaları yasaktır.
- Çok satırlı makrolar yasaktır.
- Yalnızca makro adları büyük harfle yazılır
- #if, #ifdef veya #ifndef'ten sonra gelen karakterleri girintili yapmalısınız

II.8 Yasak Şeyler!

Zorunlu kısım

- Şunları kullanmanıza izin verilmiyor:
  - for (yüzüne düştüğü için)
  - do...while
  - switch
  - case
  - goto
- İç içe üçlü operatörler “Bool ? True : False”

II.9 Yorumlar

Zorunlu kısım

- Yorumlar tüm kaynak dosyalarda bulunabilir.
- Fonksiyonların gövdesinde herhangi bir yorum olmamalıdır.
- Yorumlar tek bir satırla başlar ve biter. Tüm satırlar aracılar onlarla sıraya girer ve '**' ile başlar.
- // içinde yorum yok.

Önerilen parça

- Yorumlarınız İngilizce ve faydalı olmalıdır.
- Yorum, bir piç işlevini haklı çıkaramaz.

II.10 Dosya

Zorunlu kısım

- Bir .c ekleyemezsiniz.
- Bir .c dosyasında 5'ten fazla fonksiyon tanımınız olamaz.

II.11 Makefile

Önerilen parça

- $ (NAME), clean, fclean, re ve tüm kurallar gereklidir.
- Makefile yeniden bağlanırsa proje işlevsel değil olarak kabul edilir.
- Çoklu ikili proje durumunda, önceki kurallara ek olarak, iki ikiliyi derleyen bir tüm kuralın yanı sıra her biri için özel bir kurala sahip olun derlenmiş ikili.
- İşlev kitaplığı kullanan bir proje olması durumunda (örneğin bir libft), makefile dosyanız bu kitaplığı otomatik olarak derlemelidir.
- "Joker karakter" (örneğin * .c) kullanımı yasaktır.