

Bitirme Çalışması

Yazar

İçindekiler

1. Proje Adı	1
2. Amaç	3
Sözlük	5
3. Metin	7
3.1. Satır Sonunu Sabitleme	7
3.2. Satır sonunu koruma	8
3.3. AsciiDoc'da Metin Biçimlendirme	8
3.3.1. Kalın ve italik metin	8
3.3.2. Örnekler	8
3.3.3. Altı ve üstü çizili metin	8
3.3.4. Örnekler	9
4. Görüntü Ekleme	11
4.1. Örnek	11
4.2. Örnek	12
4.3. Şekil Başlıklarını Özelleştirme(Customize the Figure Captions)	13
4.4. Figure caption	13
5. Tablo Ekleme	15
5.1. Örnekler	15
5.2. Örnekler	15
5.3. Örnekler	16
5.4. Örnekler	16
6. Kod Ekleme	17
6.1. Kaynak Kod	18
6.1.1. Başlıklı listeleme bloğu	18
6.1.2. Başlık ve sözdizimi vurgulamalı kod bloğu	18
6.2. Kaynak Vurgulamayı Etkinleştirme	19
6.3. Kaynak Vurgulama	19
6.3.1. Belirtme çizgileri olan kod bloğu	19
6.3.2. Seçilemeyen bilgileri içeren kod bloğu	20
6.3.3. Seçilemeyen bir belirtme çizgisine sahip XML kod bloğu	20
6.3.4. Dosyadan kaynaklı kod bloğu	20
6.3.5. Kaynak dizine göre dosyadan kaynaklı kod bloğu	21
6.3.6. Kaynaktan gelen satır arası kod bloğu	21
6.3.7. Sınırlayıcı içermeyen kod bloğu	21

7. Kaynakça	23
-------------------	----

1

Proje Adı

2

Amaç

Proje Amacı...

Sözlük

Terim-I	Karşılık gelen tanım.
Terim-II	Karşılık gelen tanım.

3

Metin

Paragraflar AsciiDoc'ta herhangi bir özel biçimlendirme gerektirmez.

Bir paragraf ardışık metnin yalnızca bir veya daha fazla satırıdır.

Yeni bir paragrafa başlamak için, en az bir boş satır ile ayırın. Paragraf içindeki yeni satırlar görüntülenmez.

3.1. Satır Sonunu Sabitleme

Alt alta yazılmış cümleler bir paragrafta birleştirileceğinden cümleleri ayrı bir satıra koymak için satır sonuna bir boşluk ardından artı işareti (+) eklenmelidir. Bir diğer seçenek ise paragraftan önce hardbreaks eklenmesidir.

```
Kar kırmızıdır. +  
Göl beyaz
```

Kar kırmızıdır

Göl beyaz

```
[%hardbreaks]  
Yağmur ıslaktır.  
Güneş sıcak.
```

Yağmur ıslaktır.

Güneş sıcak.

Belgenin tamamında satır sonlarını korumak için, belgenin başlığına hardbreaks niteliğini eklenmelidir.

```
= Satır sonunu koruma  
:hardbreaks:
```

```
Bugün kar yağdı.  
Yarın güneş açacak.
```

3.2. Satır sonunu koruma

Bugün kar yağdı.
Yarın güneş açacak.

3.3. AsciiDoc'da Metin Biçimlendirme

Tıpkı konuştuğumuzda bazı kelimeleri ve cümleleri vurguladığımız gibi, metinleri biçimlendirme ile vurgulayabiliriz. Kalın veya tek aralık gibi bu biçimlendirme, çevreleyen harfler, kelimeler veya basit işaretleme içeren ifadelerle belirtilir.

3.3.1. Kalın ve italik metin

Bir sözcüğü vurgulamanın en yaygın iki yolu, onu koyu veya italik olarak biçimlendirmektir.

Bir kelimeye veya cümleye kalın stil uygulamak için, biçimlendirmek istediğiniz metnin başına ve sonuna bir yıldız işareti (*) yerleştirin. Bir harf veya harfleri bir metin dizisinde kalınlaştırmak için, harf veya harflerden önce ve sonra iki yıldız işareti (**) yerleştirin .

Harfler ve kelimeler bir (_) veya iki (__) alt çizgi kullanılarak italikleştirilir . Bir harf veya sözcüğü kalınlaştırmak ve italikleştirmek istediğinizde, kalın işaretleme en dıştaki işaretleme olmalıdır.

3.3.2. Örnekler

Elma **Eski Türkçe'de alma** diye bilinen adının, *meyvenin rengi olan al* (kırmızı)'dan geldiği **bilinmektedir**.

3.3.3. Altı ve üstü çizili metin

Bir kelime veya cümleyi altı çizili göstermek için kelime veya cümlenin başına ([underline]#) sonuna ise (#) işaretlerini yerleştiririz.

Bir kelime veya cümleyi üstü çizili göstermek için kelime veya cümlenin başına ([line-through]#) sonuna ise (#) işaretlerini yerleştiririz.

3.3.4. Örnekler

AsciiDoc öğrenilmesi kolay olduğu kadar zor bir uygulamadır.

4

Görüntü Ekleme

Görüntüler, varsayılan olarak boş olan **images::** belge özneliliğinin değerine göre çözümlenir . **images::** her görüntü makrosunda resimlerinize ortak yolu zor kodlamaktan kaçınmak için bu özelliği kullanmanız teşvik edilir .

images:: nitelik mutlak yolu görelili yolu veya temel URL olabilir. Resim hedefi bir URL veya mutlak bir yol olduğunda, imagesdir öneki hazır değildir.

- Bir görüntüyü kendi satırına dahil etmek için (yani, bir blok görüntü), **image::** öndeki dosya adını ve köşeli parantezleri kullanın.

4.1. Örnek



- Alt metni belirtmek istiyorsanız, köşeli parantezin içine ekleyin:

4.2. Örnek



- Ayrıca görüntüye bir kimlik, bir başlık verebilir, boyutlarını ayarlayabilir ve bir bağlantı oluşturabilirsiniz.

1. [#img-sunset]
2. .Bir dağ gün batımı
3. [link=https://www.flickr.com/photos/javh/5448336655]
4. image::images/sunset.png[Sunset,300,200]

1→Bloğa bir kimlik atar; [bkz bir Anchor tanımlama¹](#) .

2→Oluşturulduğunda görüntünün altında görüntülenecek blok görüntünün başlığını tanımlar.

3→link görüntüyü bir bağlantı yapar (bu, makro bloğun öznitelik listesinde de tanımlanabilir)

4→İlk konumsal nitelik olan Sunset , görüntünün alt metnidir.

İkinci ve üçüncü konumsal nitelikler sırasıyla genişlik ve yüksekliği tanımlar.

Sonuç aşağıdaki gibi görünüyor.



Şekil 4.1. Bir dağ gün batımı

¹ <https://asciidoctor.org/docs/user-manual/#images>

4.3. Şekil Başlıklarını Özelleştirme(Customize the Figure Captions)

Asciidoctor ile belgelerimizdeki görüntüleri **image** direktifi ile kullanabiliriz. Belge dönüştürüldüğünde, her resim bir resim yazısı alır. Varsayılan olarak resim yazısı etiketi, *Figure* görüntünün belgedeki konumu için bir sayı alır. Bu yüzden, ilk görüntünün *Figure 1..* şeklinde bir başlığı vardır. Görüntüye bir blok başlığı eklersek (. ile eklenmiş metin), o zaman da başlıkta aynı metin kullanılır. Resim yazısı etiketini, resim sayacını, resim yazı metnini özelleştirebilir veya belge ve resim niteliklerinin bir birleşimini kullanarak resim yazılarını devre dışı bırakabiliriz.

Aşağıdaki Asciidoctor işaretlemesine sahibiz. Birkaç resim ekliyoruz ve resim yazısı ayarlarını özelleştiriyoruz. Resim yazısı etiketini (Figure) değiştirmek için, resim yazısı (figure caption) belgesi niteliği için farklı bir değer belirledik. Örneğimizde **Logo** değerini kullanıyoruz. Bu tanımlamayı izleyen tüm başlıklar, Logo etiketine sahip olacaktır.

Ayrı bir sayaç kullanmak için, resmimizin başlık özneliği içindeki bir sayaç özelliğini kullanabiliriz. Örneğimizde bunu **Gradle** logosu için kullanıyoruz. Blok başlığı yerine başka bir başlık metni kullanmak istiyorsak, bir görüntü için title özelliğini kullanabiliriz.

Son olarak, tüm resim başlıklarını devre dışı bırakmak için resim başlığı belge özelliğini yok sayıyoruz.

4.4. Figure caption



Şekil 4.2. SDKMAN!



Şekil 4.3. Groovy



Şekil 4.4. Gradle



Şekil 4.5. Ratpack library



Şekil 4.6. Grails

5

Tablo Ekleme

Asciidoctor'da bir tablo tanımladığımız zaman, sütunların tümü aynı genişliğe sahiptir ve tablo sayfanın tüm genişliğini gösterir. Tabiki bunu tanımladığımız zaman bunu değiştirebiliriz. Tablonun sütun sayısını **column** niteliği ile değiştirebiliriz . Satır sayısını **rows** özniteliği ile belirtiriz .

Aşağıdaki sözdizimini kullanarak AsciiDoc'ta tablolar da oluşturabilirsiniz.

5.1. Örnekler

Tablo 5.1. Bu tablonun isteğe bağlı başlığı

Sütun 1'in Adı	Sütun 2'nin Adı
Sütun 1'deki hücre, satır 1	Sütun 2'deki hücre, satır 1
Sütun 1'deki hücre, satır 2	Sütun 2'deki hücre, satır 2

Tablo oluştururken kullanım amacına göre şekil ayarı verebiliriz.Örneğin;

5.2. Örnekler

Tablo 5.2. Örnek bir tablo 1

0	a	4
1	b	3
2	c	2
3	d	1
4	e	0

5.3. Örnekler

Tablo 5.3. Örnek bir tablo 2

1	2	3	4
a	b	c	d
A	B	C	D

5.4. Örnekler

Tablo 5.4. Örnek bir tablo 3

Tarih	Saat	Kontenjan	Ders
22-Aug-08	10:24	157	Bilgisayar Mühendisliğine Giriş(1. Sınıflar için)
22-Aug-08		152	Diferansiyel Denklemler(2.Sınıflar için)
24-Aug-08	TATİL		

6

Kod Ekleme

Geliştiriciler, kodun yapısını vurgulamak için renklendirilmiş kaynak kodunu görmeye alışıktır (anahtar kelimeler, türler, sınırlayıcılar vb.). Bu teknik *sözdizimi vurgulama* olarak bilinir . Bu teknik çok yaygın olduğu için AsciiDoctor, belgenizdeki kaynak kod bloklarını vurgulamak için sözdizimi için zengin kütüphaneleri bir araya getirir. Entegre kütüphanelerin listesi Rouge, CodeRay, Pygments, highlight.js ve güzelleştirmeyi içerir.

6.1. Kaynak Kod

6.1.1. Başlıklı listeleme bloğu

Başlıklı listeleme bloğu, sözdizimi vurgulama yok

Gemfile.lock.

```
GEM
  remote: https://rubygems.org/
  specs:
    asciidoctor (1.5.6.1)

PLATFORMS
  ruby

DEPENDENCIES
  asciidoctor (~> 1.5.6.1)
```

6.1.2. Başlık ve sözdizimi vurgulamalı kod bloğu

Başlık ve sözdizimi vurgulamalı kod bloğu

app.rb.

```
require 'sinatra'

get '/hi' do
  "Hello world!"
end
```

6.2. Kaynak Vurgulamayı Etkinleştirme

Etkinleştirildiğinde, sözdizimi vurgulama, **source** blok stili ve kaynak dili olan listeleme veya değişmez bloklara uygulanır . Ancak, sözdizimi vurgulama varsayılan olarak etkin değildir .

Bir belgede sözdizimi vurgulamasını etkinleştirmek için **source-highlighter** belge niteliğini ayarlamanız gerekir . Bu niteliği belgede veya CLI veya API'den ayarlayabilirsiniz.

Özelliği belgede ayarlarsanız , **belge başlığında**¹ tanımlanması gerekir .

```
= Document Title
:source-highlighter: <value>
```

6.3. Kaynak Vurgulama

Kaynak koduna vurgulama uygulamak için, **source** blok stilini bir listeleme bloğuna, değişmez bloğa veya paragrafa eklemeniz ve bir kaynak dil belirtmeniz gerekir.

6.3.1. Belirtme çizgileri olan kod bloğu

Belirtme çizgileri olan kod bloğu

```
require 'sinatra' ❶

get '/hi' do ❷
  "Hello world!" ❸
end
```

- ❶ Kütüphane içe aktarma
- ❷ URL eşleme
- ❸ HTTP yanıt gövdesi

¹ <https://asciidoctor.org/docs/user-manual/#doc-header>

6.3.2. Seçilemeyen bilgileri içeren kod bloğu

Seçilemeyen bilgileri içeren kod bloğu

```
----  
kod satırı ❶  
kod satırı ❷  
kod satırı ❸  
----
```

- ❶ C tarzı diller için bir çizgi yorumunun arkasındaki belirtme çizgisi.
- ❷ Ruby, Python, Perl, vb. için çizgi yorumunun arkasındaki belirtme çizgisi.
- ❸ Clojure için bir çizgi yorum arkasındaki belirtme çizgisi.

6.3.3. Seçilemeyen bir belirtme çizgisine sahip XML kod bloğu

Seçilemeyen bir belirtme çizgisine sahip XML kod bloğu

```
<section>  
  <title>Bölüm Başlığı</title> ❶  
</section>
```

- ❶ Bölüm başlığı gerekli.

6.3.4. Dosyadan kaynaklı kod bloğu

Kaynak Kodu dahil eden kod bloğu.

```
[source, ruby]  
----  
include::app.rb[]  
----
```


6.3.5. Kaynak dizine göre dosyadan kaynaklı kod bloğu

Kaynak dizine göre dosyadan kaynaklı kod bloğu.

```
:sourcedir: src/main/java
[source,java]
----
include::{sourcedir}/org/asciidoctor/Asciidoctor.java[]
----
```

6.3.6. Kaynaktan gelen satır arası kod bloğu

Kaynaktan gelen satır arası kod bloğu.

```
[source,ruby,indent=0]
----
include::lib/document.rb[lines=5..10]
----
```



indent 0 olduğunda, satır bloğu girintisi sıyrılır (sekmeler 4 boşlukla değiştirilir).

indent > 0 olduğunda, öncü blok girintisi önce sıyrılır (sekmeler 4 boşlukla değiştirilir), sonra bir blok bu değere eşit sütun sayısı ile girintili olur.

6.3.7. Sınırlayıcı içermeyen kod bloğu

Sınırlayıcı içermeyen kod bloğu (boş satır yok)

```
<meta name="viewport"
  content="width=device-width, initial-scale=1.0">
Bu normal bir içeriktir.
```


Kaynakça

- [1] İNNER, A. B., & KÜÇÜK, S. Genel Stewart Platformları (GSP) için Boyutsal Eniyileme Yazılımının ve Yeniden Ayarlanabilir 3 Bacaklı bir GSP Mekanizmasının Geliştirilmesi.
- [2] EKİM, H. E., & İNNER, A. B. (2021). DUYGU ANALİZİ VE FİKİR MADENCİLİĞİ UYGULAMALARI ÜZERİNE LİTERATÜR TARAMASI. Kahramanmaraş Sütçü İmam Üniversitesi Mühendislik Bilimleri Dergisi, 24(2), 93-114.
- [3] İNNER, B., Küçük, S., & Bingül, Z. (2010). 'Farklı yapıdaki stewart platformlarının tek bir çatı üzerinden tasarımı ve benzetimi. Otomatik Kontrol Ulusal Toplantısı, Sivas.

Websites

[Official Node.js Website](http://nodejs.org/)¹

[Latest Node.js documentation](http://nodejs.org/docs/latest/api/index.html)²

[Node.js Style Guide](http://nodeguide.com/style.html)³

[Node.js Tutorial](http://www.nodebeginner.org/)⁴

¹ <http://nodejs.org/>

² <http://nodejs.org/docs/latest/api/index.html>

³ <http://nodeguide.com/style.html>

⁴ <http://www.nodebeginner.org/>

