

# Vaadin UI Layouts

# Layout Bileşenleri

First Name Kenan
Last Name Sevindik
<b>Add</b>

→ VerticalLayout

First Name	Last Name	<b>Add</b>

→ HorizontalLayout

First Name	Last Name
<b>Add</b>	

→ GridLayout(2,2)

A Field *	
Another Field !	

→ FormLayout

This is the error indicator of a Field.

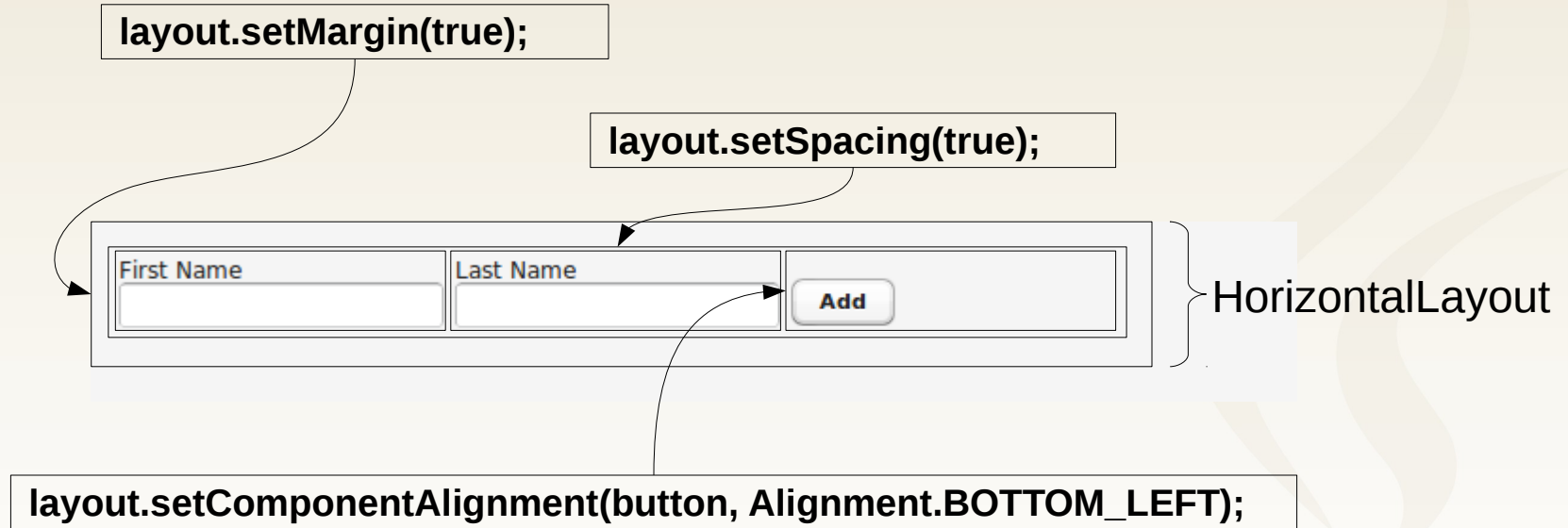
# Layout Bileşenlerinin Ortak Özellikleri

- **setSpacing(true)** metodu kullanılarak layout'un içine yerleştirilen bileşenler arasında boşluk bırakılması sağlanabilir
  - Bırakılacak boşluk miktarı **CSS stilleri** ile ayarlanabilir
- Layoutların çevrelerinde boşluk bırakmak için ise **setMargin()** metodu kullanılabilir
  - `layout.setMargin(true);`
  - `layout.setMargin(new MarginInfo(false, true, false, true));`

# Layout Bileşenlerinin Ortak Özellikleri

- Bileşenlerin layout içerisindeki konumları da **setComponentAlignment()** metodu kullanılarak belirtilebilir
  - `layout.setComponentAlignment(component, Alignment.BOTTOM_LEFT);`
- Her bir bileşen için layout içinde ne kadar alan ayrılacağı **setExpandRatio()** metodu ile belirlenebilir

# Layout Bileşenlerinin Ortak Özellikleri



TOP_LEFT	TOP_CENTER	TOP_RIGHT
MIDDLE_LEFT	MIDDLE_CENTER	MIDDLE_RIGHT
BOTTOM_LEFT	BOTTOM_CENTER	BOTTOM_RIGHT

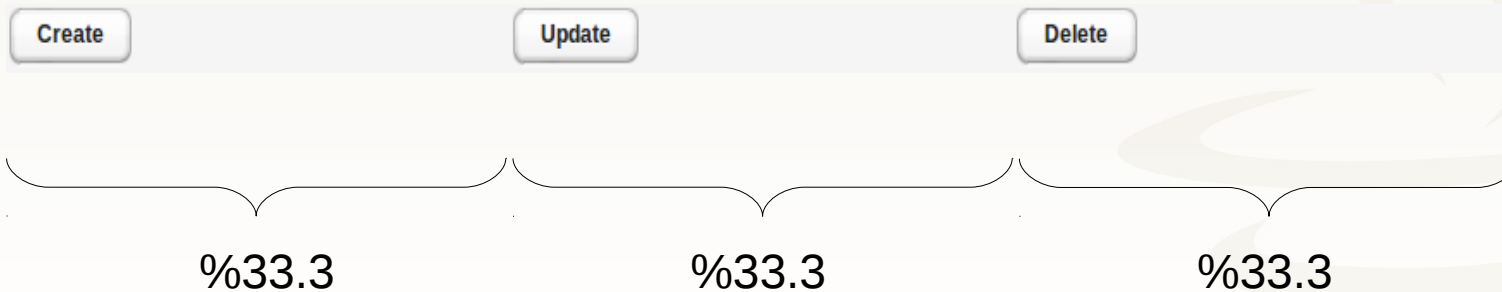
# Layout Expand Ratio

```
HorizontalLayout layout = new HorizontalLayout();

layout.setMargin(true);
layout.setSpacing(true);
layout.setWidth("800px");

Button btn1 = new Button("Create");
Button btn2 = new Button("Update");
Button btn3 = new Button("Delete");

layout.addComponents(btn1, btn2, btn3);
```



# Layout Expand Ratio

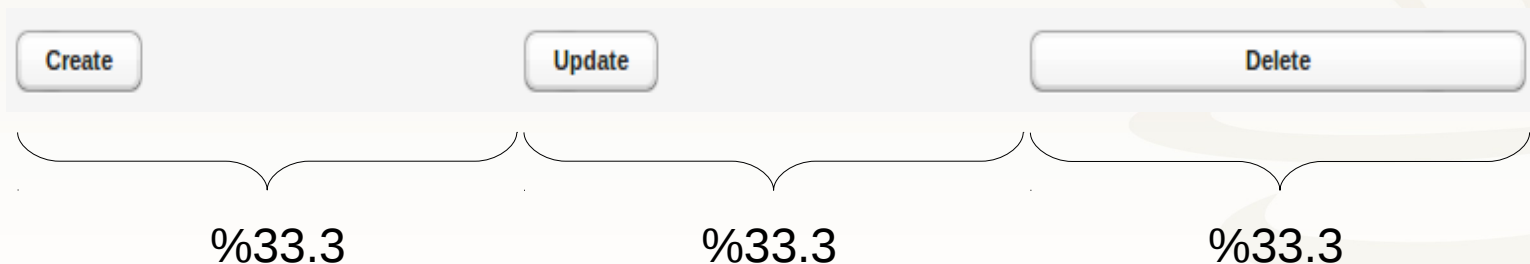
```
HorizontalLayout layout = new HorizontalLayout();

layout.setMargin(true);
layout.setSpacing(true);
layout.setWidth("800px");

Button btn1 = new Button("Create");
Button btn2 = new Button("Update");
Button btn3 = new Button("Delete");

btn3.setSizeFull();

layout.addComponents(btn1, btn2, btn3);
```



# Layout Expand Ratio

```
HorizontalLayout layout = new HorizontalLayout();

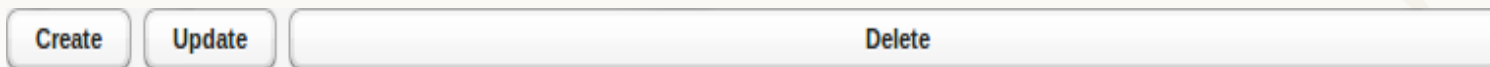
layout.setMargin(true);
layout.setSpacing(true);
layout.setWidth("800px");

Button btn1 = new Button("Create");
Button btn2 = new Button("Update");
Button btn3 = new Button("Delete");

btn3.setSizeFull();

layout.addComponents(btn1, btn2, btn3);

layout.setExpandRatio(btn3, 1.0f);
```



İki butona ihtiyaç  
duydukları minimum  
yer verildi

Geri kalan bütün alan üçüncü  
butona bırakıldı



# Layout Expand Ratio

```
HorizontalLayout layout = new HorizontalLayout();
```

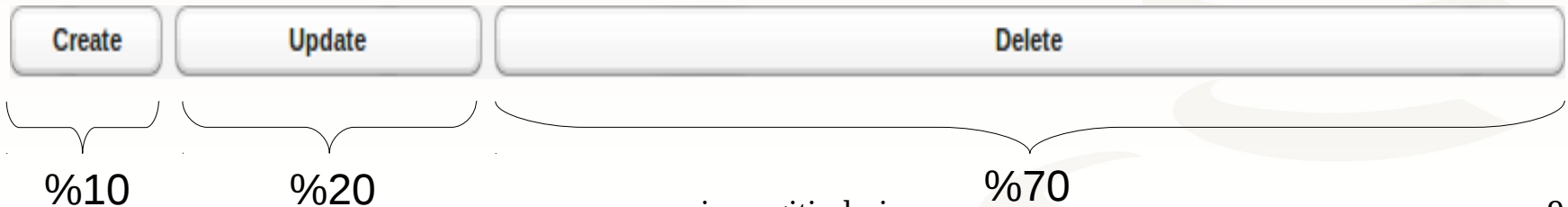
```
layout.setMargin(true);  
layout.setSpacing(true);  
layout.setWidth("800px");
```

```
Button btn1 = new Button("Create");  
Button btn2 = new Button("Update");  
Button btn3 = new Button("Delete");
```

```
btn1.setSizeFull();  
btn2.setSizeFull();  
btn3.setSizeFull();
```

```
layout.addComponents(btn1, btn2, btn3);
```

```
layout.setExpandRatio(btn1, 0.1f);  
layout.setExpandRatio(btn2, 0.2f);  
layout.setExpandRatio(btn3, 0.7f);
```



# VerticalLayout

- Bileşenlerin dikeyde alt alta sıralı biçimde yerleştirilmelerini sağlar
- Default **yüksekliği tanımsızdır** (undefined)
- Dolayısı ile eklenen bileşen sayısına göre genişler
- Default **genişliği ise %100'dür**
- Dolayısı ile yatayda kendine ayrılmış bütün alanı kaplar

# HorizontalLayout

- Bileşenlerin yatayda yan yana sıralı biçimde yerleştirilmelerini sağlar
- Default **genişliği tanımsızdır** (undefined)
- Dolayısı ile eklenen bileşen sayısına göre genişler
- Default **yüksekliği de tanımsızdır** (undefined)

# GridLayout

- **Satır ve sütun sayısı** belirli bir **grid** sunar
- Bileşenler bu grid içerisinde **x ve y koordinatları** ile belirlenen kısımlara yerleştirilebilirler
- Bir bileşen grid içerisinde **birden çok hücreyi** kaplayabilir

R/C 1	Col 1	Col 2	Col 3
Row 1	3x1 button		
Row 2	1x2 cell	A 2x2 date field	
		<div><div>◀◀</div><div>◀</div><div>November</div><div>▶</div><div>▶▶</div></div> <div>2007</div> <div><div>Sun</div><div>Mon</div><div>Tue</div><div>Wed</div><div>Thu</div><div>Fri</div><div>Sat</div></div> <div><div></div><div></div><div></div><div></div><div>1</div><div>2</div><div>3</div></div>	
Row 3		<div><div>4</div><div>5</div><div>6</div><div>7</div><div>8</div><div>9</div><div>10</div></div> <div><div>11</div><div>12</div><div>13</div><div>14</div><div>15</div><div>16</div><div>17</div></div> <div><div>18</div><div>19</div><div>20</div><div>21</div><div>22</div><div>23</div><div>24</div></div> <div><div>25</div><div>26</div><div>27</div><div>28</div><div>29</div><div>30</div></div>	

# GridLayout

```
TextField firstName = new TextField("First Name");
TextField lastName = new TextField("Last Name");
TextArea address = new TextArea("Address");
ComboBox city = new ComboBox("City",
    PetClinicService.getInstance().getCities());
TextField telephone = new TextField("Telephone");
OptionGroup phoneType = new OptionGroup("Phone Type",
    Arrays.asList(PhoneType.values()));
```

```
GridLayout layout = new GridLayout();
```

```
layout.setColumns(2);
layout.setRows(4);
```

```
layout.addComponent(firstName, 0, 0);
layout.addComponent(lastName);
layout.addComponent(address, 0, 1, 0, 3);
layout.addComponent(city);
layout.addComponent(telephone);
layout.addComponent(phoneType);
```

Başlangıç  
hücresinin  
koordinatı

Bitiş  
hücresinin  
koordinatı

Sıradaki boş hücreye

0	First Name *	Last Name *
1	Address *	City *
2		Telephone *
3		Phone Type * <input type="radio"/> Home <input type="radio"/> Work

# FormLayout

- **Sol tarafta** bileşenlerin **başlıkları**, **sağ tarafta** da **bileşenlerin** kendileri olacak biçimde iki sütun halinde bileşenleri yukarıdan aşağıya sıralar
- **VerticalLayout**'a benzer
- Required alanlar için ve hatalı girdiler için kullanılan uyarılar **başlığın yan kısmında** görüntülenir
- Bir **Form** UI bileşeni ile beraber kullanılır

# AbsoluteLayout

- **AbsoluteLayout**, içermekte olduğu bileşenleri ekranda yukarı-aşağı, sağ-sol ya da z-index (içeri-dışarı) değerlerine göre konumlandırır
- Bu layout içinde **CSS özelliklerine** doğrudan müdahale edildiği için **tarayıcı uyumluluğuna** ayrıca dikkat edilmelidir

```
AbsoluteLayout layout = new AbsoluteLayout();  
layout.setWidth("400px");  
layout.setHeight("250px");  
  
TextField text = new TextField("Hello World");  
layout.addComponent(text, "left: 50px; top: 50px;");
```

# CustomLayout

- CustomLayout, bileşenlerin yerleşimini **HTML** kullanarak yönetmeye izin verir
- CustomLayout kullanıldığında java kodunun yanı sıra bir de **HTML dosyası** oluşturulmalıdır

```
<table width="100%" height="100%">
  <tr height="100%">
    <td>
      <table align="center">
        <tr>
          <td align="right">User&nbsp; name:</td>
          <td><div location="username"></div></td>
        </tr>
        <tr>
          <td align="right">Password:</td>
          <td><div location="password"></div></td>
        </tr>
      </table>
    </td>
  </tr>
  <tr>
    <td align="right" colspan="2">
      <div location="okbutton"></div>
    </td>
  </tr>
</table>
```

webapp/VAADIN/themes/  
mytheme/layouts/  
mylayout.html

```
CustomLayout custom = new CustomLayout("mylayout");
```

```
TextField username = new TextField();
custom.addComponent(username, "username");
```

```
TextField password = new TextField();
custom.addComponent(password, "password");
```

```
Button ok = new Button("Login");
custom.addComponent(ok, "okbutton");
```



# Layout'ların İç içe Kullanımı

En dıştaki layout UI content'ine atanır

```
@Override
protected void init(VaadinRequest request) {

    VerticalLayout layout = new VerticalLayout();
    layout.setMargin(true);
    setContent(layout);

    FormLayout generalInfoLayout = new FormLayout();
    layout.addComponent(generalInfoLayout);

    //generalInfoLayout icerigi
    ...

    GridLayout addressLayout = new GridLayout(2,3);
    addressLayout.setSpacing(true);
    layout.addComponent(addressLayout);

    //addressLayout icerigi
    ...
}
```

The diagram illustrates a user form layout. At the top, there are two input fields for 'First Name' and 'Last Name'. Below these, the form is divided into two main sections. The left section is titled 'Address' and contains a large text area for the address and an 'Add' button at the bottom. The right section is titled 'City' and contains a dropdown menu, two radio buttons for 'Phone Type' (HOME and WORK), and a 'Telephone' input field. Arrows from the text boxes point to the corresponding components in the form: 'En dıştaki layout UI content'ine atanır' points to the outer container, 'Layout'lar birbiri altına eklenebilir' points to the 'generalInfoLayout' and 'addressLayout' components, and another arrow points to the 'Add' button.

Layout'lar birbiri altına eklenebilir

- Harezmi Bilişim Çözümleri
- Kurumsal Java Eğitimleri
- <http://www.java-egitimleri.com>
- [info@java-egitimleri.com](mailto:info@java-egitimleri.com)

