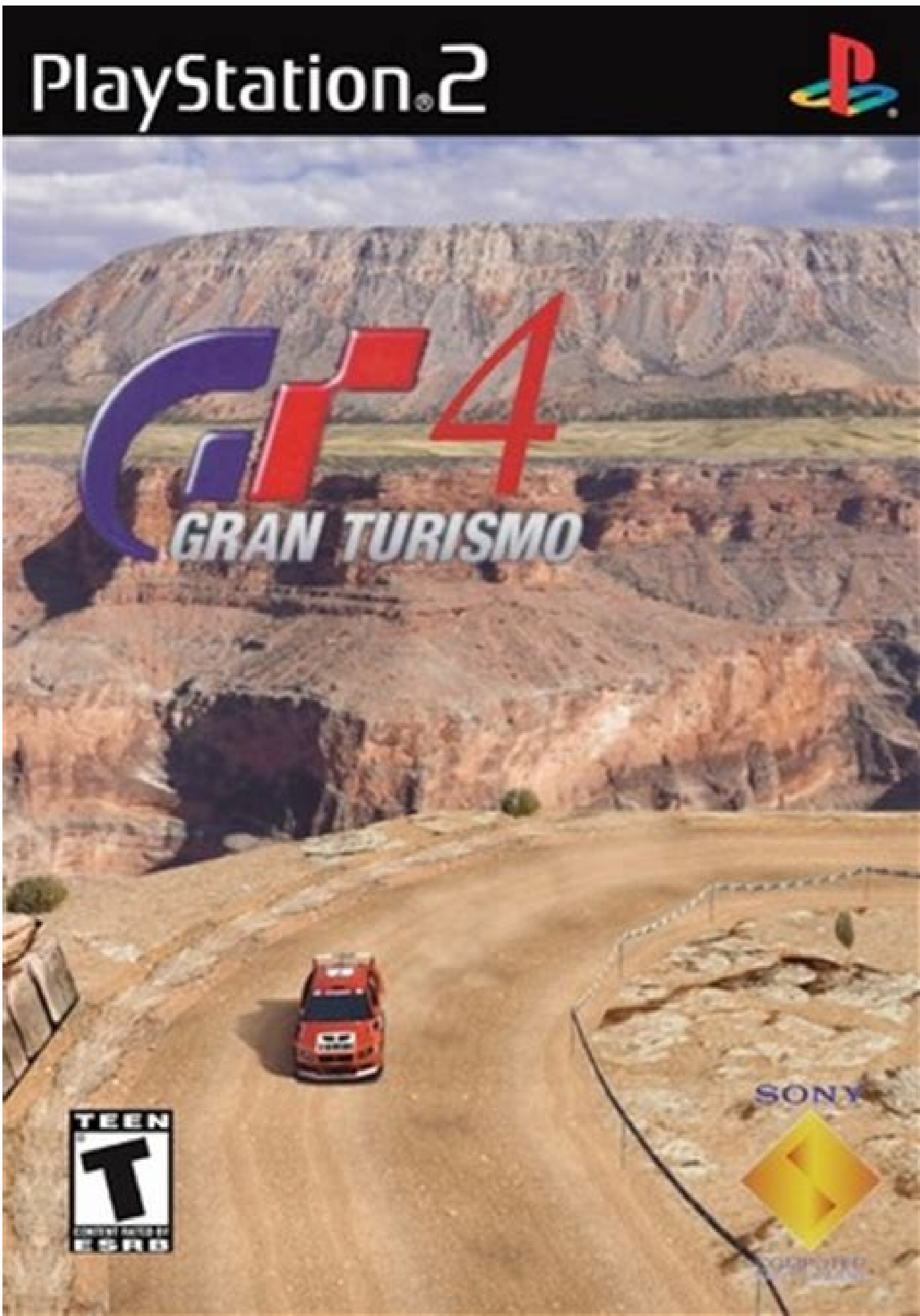


Input box in android

Continue





How to add input boxes in html. Input text box in android studio. How to add checkbox in input box. Dialog box in android with text input example. Input box in android studio. How to input box in html.

Accept Valid only for the input file type, the Accept attribute specifies which file types can be selected in the file upload control. View the input file type. alt Valid only for an image button, the alt attribute provides alternative text for the image and displays the attribute's value when the image source is missing or otherwise unloading. Show image input type. autocomplete (not a boolean attribute!) The autocomplete attribute takes as its value a space-delimited string that describes what type of autocomplete should return input. A typical autocomplete implementation is reminiscent of previous values entered into the same input field, but more complex autocomplete forms can exist. For example, the browser can integrate your device's contact list to automatically enter email addresses in the email input field. For valid values, see HTML Attribute Values: AutoComplete. Autocomplete attribute applies to Hidden, Text, Search, URL, Phone, Email, Date, Month, Week, Time, DateTime, Number, Range, Color and Password. This attribute does not affect input types that do not return numeric or text data, and applies to all input types except checkboxes, radio buttons, files, and buttons of any kind. For more information, see the HTML AutoComplete attribute, including password security information and about how auto-completion is slightly different for hidden and other input types. autofocus A boolean attribute that, if present, indicates that the input should be automatically focused when the page finishes loading (or when the containing element is displayed). Note: An element with an autofocus attribute can receive focus before the DOMContentLoaded event is dispatched. At most one element in a document can have the autofocus attribute. If placed on more than one element, the first one with the attribute gets the focus. The autofocus attribute cannot be used on hidden inputs because hidden inputs cannotWarning: The form control's autofocus can be confusing for visually impaired people who use screen reading technology and people with cognitive impairments. When autofocus is assigned, screen readers "teleport" the user to the form control without warning. Carefully consider accessibility when using the autofocus attribute. Auto focus on the control may cause the page to scroll while loading. Focus can also display dynamic keyboards on some touch devices. While a screen reader announces the focus control's label, the screen reader doesn't announce anything before the label, and a sighted user on a small device will also not notice the context created by the previous content. capture The capture attribute was introduced in the HTML Media Capture specification and applies only to the input file type. The capture attribute specifies which media attribute - microphone, video, or camera - should be used to capture a new file for upload using the file upload control in support scenarios. . View input file type. checked Valid for radio types and checkboxes, selected is a boolean attribute. If it is present for an option type, it means that the radio button is in the currently selected group of radio buttons with the same name. If specified for a checkbox type, indicates that the checkbox is checked by default (on page load). It does not indicate whether this checkbox is currently selected: if the state of the checkbox is changed, this content attribute will not reflect the change. (Only the enabled IDL attribute of the HTMLInputElement is updated.) Attention. Unlike other input controls, the values of checkboxes and option buttons are included in the submitted data only when they are currently selected. If so, the name and value(s) of the controls tested shall be provided. For example, if the Fruit checkbox is set to Cherry and this checkbox is selected, the submitted form data will contain Fruit=Cherry. If the checkbox is not active then data in the form in general. The default value for checkboxes and option buttons is on. dirname The dirname attribute applies only to text input and search types, it allows you to specify the direction of an element. When added, the form control sends two name/value pairs: the first is name and value, and the second is dirname as the name with the lr or rtl value set by the browser. Fruits: When the above form is submitted, the input results in the name/value pair fruit=cherry and the dirname/direction pair fruit.dir=ltr. disabled Boolean attribute that, if present, indicates that the user should not interact with the input. Disabled inputs are usually grayed out or otherwise indicate that the field is not available for use. Specifically, disabled input will not receive a click event, and disabled input will not be submitted with the form. form A string indicating the element to which the input is associated (ie, the owner of its form). The value of this string, if present, must match the ID of the element in the same document. If this attribute is not specified, the element is bound to the nearest containing form, if any. The form attribute allows you to place the input anywhere in the document, but connect it to a form elsewhere in the document. Note: Inputs can only be assigned to one form. formatting only applies to images and upload input types. For more information, see Recording input type. formnovalidate Applies to image and submitonly types. For more information, see Recording input type. formmethod Applies only to image type and input stream. For more information, see Recording input type. formnovalidate Applies to image and submitonly types. For more information, see Recording input type. formtarget only applies to the image type and input transport. For more information, see Recording input type. height Only applies to the input image button, height is the height of the image file to be displayed as a representation of the graphical submit button. See image input type. id Global attribute valid for all elements,

including all input types, specifies a unique identifier (ID) that must be unique throughout the document. Its purpose is to identify the element during connection. The value is used as the value of the attribute to associate with the form control. See `inputmode` A global attribute for all elements that tells browsers what type of virtual keyboard configuration to use when editing this element or its content. Values include `none`, `text`, `phone`, `URL`, `email`, `number`, `decimal`, and `search`. `list` The value of the `List` attribute must be the id of the element in the same document, provides a predefined list of values that the user can suggest for this input. Any values in the list that do not match the type are not included in the suggested options. The values provided are recommendations, not requirements: users can choose from this predefined list or enter a different value. Applies to `text`, `search`, `URL`, `phone`, `email`, `date`, `month`, `week`, `time`, `date` and `local time`, `number`, `range`, and `color`. According to the specifications, the `list` attribute is not supported by `hidden`, `password`, `checkbox`, `radio`, `file`, or any of the button types. Depending on the browser, the user may see a suggested custom color palette, full-range markers, or even an input that opens as but allows values without a list. See the browser compatibility table for other input types. See the element. `max` Applies to `date`, `month`, `week`, `time`, `date`, `time`, `number`, and `range`, specifies the largest value in the rangevalues. If the value entered in the element exceeds this value, the element will fail the constraint test. If the value of the `max` attribute is not a number, the element does not have a maximum value. There is a special case: if the data type is `periodic` (such as dates or times), the maximum value can be less than the minimum value, which means that the range can repeat; for example, it allows you to enter a time range from 22:00 to 4:00. `maxlength` Applies to `text`, `search`, `URL`, `phone number`, `email address`, and `password`, specifies the maximum number of characters (in UTF-16 code units) that a user can enter in a field. Must be an integer equal to or greater than 0. If no maximum length is specified or an invalid value is provided, the field has no maximum length. This value must also be greater than or equal to `minlength`. If the length of the text entered in the field exceeds the maximum length of UTF-16 code units, the input will fail the constraint check. By default, browsers prevent users from entering more characters than the `maxlength` attribute allows. For more information, see Client-Side Authentication. `min` Applies to `date`, `month`, `week`, `time`, `local date/time`, `number`, and `range`, and specifies the most negative value in the range of valid values. If the value entered in the element is less than this value, the element will fail the constraint test. If the value of the `min` attribute is not a number, the element does not have a minimum value. This value must be less than or equal to the value of the `max` attribute. If the `min` attribute is present but not specified or invalid, the minimum value is not used. If the `min` attribute is valid and the non-empty value is less than the minimum allowed by the `min` attribute, the constraint check will prevent the form from being submitted. For more information, see Client-Side Authentication. There is a special case: if the data type is `periodic` (such as dates or times), the maximum value can be less than the minimum value, which means that the range canaround; For example, you can specify a time range from 10:00 PM to 4:00 AM. `minlength` Applies to `text`, `search`, `url`, `tel`, `email`, and `password`, specifies the minimum number of characters (in UTF-16 code units) that the user can enter in the input field. It must be a non-negative integer less than or equal to the value specified by `maxlength`. If `minarum` is not specified or an invalid value is specified, the input has no minimum length. Input limit validation fails if the length of the text entered in the field is less than the minimum length of UTF-16 code units, preventing the form from being submitted. See Client-Side Authentication for more information. Multiple The multiple boolean attribute, if set, means that the user can enter comma-separated email addresses in the email widget or select more than one file using the file input. See email and file input method. `name` A string specifying the name of the input control. This name is sent along with the value of the control when the form data is submitted. Consider the name a required attribute (even if it isn't). If the input has no name or the name is empty, the input value will not be sent with the form! (Disabled controls, unchecked radio buttons, unchecked boxes, and reset buttons are also not sent.) There are two special cases: `charset` : when used as the name of a hidden element, the input value is automatically set by the user. character encoding agent used to submit the form. `isindex`: The name `isindex` is not allowed for historical reasons. The name attribute gives switches unique behavior. Only one toggle button in a group of toggle buttons with the same name can be activated at a time. Selecting any radio button in this group will automatically deselect all currently selected radio buttons in the same group. The value of this single checked radio button is sent along with the name when the form is submitted by traversing a series of radio button groups with the same name.Focuses when activated. If they are not grouped in their original order and one of the groups is checked, tab grouping starts when the first one in the group is encountered, skipping any unchecked tabs. In other words, if the checkbox is checked, Tab will skip the unchecked radio buttons in the group. If neither is checked, the radio button group receives focus when the first button in the group with the same name is reached. When the focus is on one of the radio buttons in a group, the arrow keys move through all radio buttons with the same name, even if the radio buttons are not grouped in their original order. When an input element is given a name, that name becomes the form element's `HTMLFormElement.elements` property. If you have one input named `quest` and another named `hatsize`, you can use the following code: `let form = document.querySelector("form"); let guestName = form.elements.guest; let hat size = form.element["hat size"]`; When this code is executed, `guestName` is the `HTMLInputElement` of the guest field, and `hatSize` is the hat size field object. Warning. Avoid naming form elements that match a built-in form property or you'll override a predefined property or method with that reference to the corresponding input. `pattern` Valid for `text`, `search`, `URL`, `phone`, `email`, and `password`. The `pattern` attribute specifies a regular expression that must match the input value in order for the value to pass the constraint check. This must be a valid JavaScript regular expression used by the `RegExp` type and described in our regular expressions guide. The `'u'` flag is specified when the regular expression is compiled, so the pattern is treated as a sequence of Unicode code points, not ASCII. The text sample must not contain a backslash. If the `model` attribute is present but not specified or invalid, noThe expression was used and completely ignored this attribute. If the `template` attribute is valid and a non-empty value does not match the template, the constraint check will prevent the form from being submitted. Note. When using a template attribute, inform the user of the expected format by adding explanatory text alongside. You can also include a header attribute that explains what requirements must be met to match the pattern. Most browsers display this name as a tooltip. The visible statement is required for accessibility. The label is an extension. For more information, see Client-Side Authentication. `placeholder` The placeholder attribute applies to `text`, `search`, `URL`, `phone`, `email`, `password`, and `number` and provides the user with a summary of what information is expected in the field. It should be a word or short phrase indicating the type of data expected, not an explanation or hint. The text must not contain line breaks or line breaks. So, for example, if the field is expected to contain the user's name and its label "Name", a suitable placeholder might be "eg. For example, Mustafa". Note. The placeholder attribute is not semantically as useful as other ways of explaining a form and can cause unexpected technical problems with your content. See Labels for more information. `readonly` A boolean attribute that, if present, specifies that the user should not be able to change the value of the input. The read-only attribute is supported by `text`, `search`, `URL`, `phone`, `email`, `date`, `month`, `week`, `time`, `datetime`, `number`, and `password` input types. For more information, see the HTML: read-only attribute. Required Required is a boolean attribute that, if present, specifies that the user must provide a value for input before submitting the owner form. The requested attribute is supported by `text`, `search`, `URL`, `phone`, `email`, `date`, `month`, `week`, `time`, `local date and time`, `number`, `password`, `checkbox`, `radio`, and `file` inputs. See Client Side AuthenticationHTML Attribute: Required for more information. `size` Applies to email address, `password`, `phone number`, `URL` and `text`. The `size` attribute determines how much of the input is displayed. This essentially produces the same result as setting the `width` property in CSS with a few specialties. The actual value unit depends on the input type. For `password` and `text` it is the number of characters (or em units) with a default value of 20, for others it is pixels (or pixel units). The CSS width takes precedence over the `size` attribute. `src` Applies only to the image submit button, `src` is a string specifying the URL of the image file to display as a representation of the graphical submit button. See image input type. `step` Applies to `date`, `month`, `week`, `time`, `local date/time`, `number`, and `range`. The `step` attribute is a number that specifies the level of detail that the value must match. If not included directly: for `number` and `range`, the step defaults to 1. Each `date/time` record type has a default step value corresponding to the type: see separate input pages: `date`, `date-time-local`, `month`, `time`, and `week`. The value must be a positive number - an integer or floating point - or the special value `any`, which means no jumps are included and any value is allowed (in addition to other restrictions such as `min` and `max`). If none is explicitly set, the valid values for input type `number`, `date/time`, and `range` are equal to the base step value - the minimum value, and the step value increases to the maximum value if specified. For example, if you have , then any even integer of 10 or greater is valid. If is omitted, any integer is valid, but floating-point numbers (such as 4.2) are not valid because the operation defaults to 1. For 4.2 to be valid, the step must be set to any , 0.1 , 0.2 or whatever, the minimum value must be a number ending in 0.2, for example Note. If the information entered by the user does not match the step configuration , the value isis invalid when checking the constraint and matches the pseudo-class `:invalid`. For more information, see Client-Side Validation. `tabindex` A global attribute valid for all elements, including all input types, an integer attribute indicating whether the element can be input-oriented (set) or should participate in sequential keyboard navigation. Since all but hidden input types can be focused, this attribute should not be used on form controls as it would require controlling the focus order for all elements in the document, which if done incorrectly could reduce usability and accessibility. `title` A global attribute that applies to all elements, including all input types, that contains text that provides help information about the element to which it belongs. Such information may typically, but not necessarily, be provided to the user in the form of a prompt. The title should NOT be used as the primary explanation of the form control's purpose. Instead, use a element with a `for` attribute that matches the form control's id attribute. See labels below. `type` A string specifying the type of control to display. For example, to create a check box, the value of the check box is used. If omitted (or given an unknown value), the text input type is used, creating a plain text input field. Valid values are listed above in the Input Types section. `value` The value of the input control. If specified in the HTML, it is the initial value and can be changed at any time from then on, or retrieved using JavaScript to access the `value` property of the corresponding `HTMLInputElement` object. The `value` attribute is always optional, but should be considered mandatory for checkboxes, radios, and hidden elements. `Width` Only valid for an image input button. `Width` is the width of the image file that will be displayed as the graphical representation of the Submit button. Show image input type. `Type` of

